DRAPED CLOTHES BASED HUMAN POSE RECOGNITION USING DEPTH IMAGES

BY

FAISAL SAJJAD

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

MARCH 2018

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **FAISAL SAJJAD** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

Dr. Khalid Al-Jasser
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

25/9/18
Date

Thesis Committee

Dr. Adel Fadhl Ahmed (Advisor)

Dr. Moataz Ahmed (Member)

Dr. Lahouari Ghouti (Member)

©Faisal Sajjad
2018

Dedicated to Almighty Allah and my parents for their continuous support and prayers for my success

ACKNOWLEDGMENTS

Foremost, I would like to express thanks to my advisor Dr. Adel Fadhl Ahmed for his guidance,continuous support and excellent cooperation which helped me a lot to achive my objectives in this thesis work. In addition, I would also like to thank my thesis committee members : Dr Moataz Ahmed and Dr. Lahouari Ghouti , for their valueable recommendations, encouragement and perceptive comments during my thesis work.

I'm highly thankful to department staff for providing me experimental environment.

A very special thank to Mr.Tahir Mohamed, IT Manager at Haward Technology Middle east , UAE and my rest of the IT department colleagues for their phenomenal support and encourgment whcih helped me to avail the scholarship in KFUPM.

I'm very grateful to Pakistani community here in KFUPM for their kind support. Also thanks to M. Jalal khan , Jawad Javed, Sultan Anwar and Zawar Butt for all the fun we have had throught my MS.

Finally, I would like to thank my Parents, brothers and sisters for their continuous support and prayers for my success.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BOF	Bag of Features
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
DPM	Deformable Part Multiscale Model
SVM	Support Vector Machine
HMM	Hidden Markov Model
MDF	Median Depth Feature
LBP-DDF	Local Binary Pattern Decimal Depth Feature
DOG	Difference of Gaussian
KNN	K Nearest Neighbour
SVD	Singular Value Decomposition
B_Hog/BOF_HOG	Bag of feature using Histogram of Oriented Gradients
B_SIFT/BOF_SIFT	Bag of feature using Scale Invariant Feature Transform
H_Avg	Histogram of Oriented Gradients Average
LF	Local Features
RF	Random Forest

THESIS ABSTRACT

NAME: Faisal Sajjad

TITLE OF STUDY: DRAPED CLOTHES BASED HUMAN POSE RECOGNITION USING DEPTH IMAGES

MAJOR FIELD: Computer Science

DATE OF DEGREE: March 2018

Human pose recognition is considered a well-known process for estimating the human body pose from a single image or a series of video frames. There exist many applications that can benefit from human pose technology e.g. activity recognition, human tracking, 3D gaming, character animation, clinical analysis of human gait and other HCI applications. Due to its many challenges, such as illumination, occlusion, outdoor environment and clothing, it is considered one of the active areas in computer vision now a days.

For the last 15 years, human pose recognition problem significantly gained interest of many researchers and therefore, many techniques were proposed in order to address the challenges of human pose recognition. So far most of the human pose recognition work is done on Western clothes where human body parts are not

covered completely in a single piece fabric. However, the recognition of human body parts in western clothes is comparably easier than draped based fabric where all the body parts are covered in single piece of fabric.

Therefore in this thesis we primarily targeted the draped based clothes especially Arabic dress. The significance of Arabic dress that it is covered in a single piece of fabric. We developed a framework for recognizing human pose in draped based clothes. In this framework we adopted learning based technique in Pixel and Patch based methods for recognizing human pose. In each method we applied depth and computer vision feature extraction techniques. These features give us a little information about a human body part. In order to get full information or prediction about each body part, the classification is performed on these features. We also used two new depth features for estimating human pose in depth images. Results show that our draped clothes framework figure it out that in pixel based approach the SIFT technique outclass other feature techniques leading with 64% accuracy. While in patch based method we found that local feature is very useful for predicting the correct body parts with almost 65% accuracy.

ملخص الرسالة

الاسم الكامل: فيصل سجاد

عنوان الرسالة: التعرف على الانسان المكتسي ثوب يغطي كامل الجسم باستخدام الصور العميقة.

التخصص: علوم حاسوب.

تاريخ الدرجة العلمية: مارس 2018م

تعتبر عملية التعرف على شكل جسم الانسان من العمليات الشائعة في علم الحاسب وذلك من خلال صورة أو فيديو يحتوي على مجموعة من المشاهد. كما أن هناك العديد من التطبيقات التي تستخدم وتستفيد من تقنية التعرف على شكل جسم الانسان مثل تتبع الانسان، الرسوم المتحركة، الألعاب ثلاثية الأبعاد، التحليل السريري طريقة مشي الانسان بالإضافة للعديد من التطبيقات التي تعتمد على تقنية تفاعل الانسان والحاسوب وغيرها الكثير. تعد علمية التعرف على شكل جسم الانسان جزء من المجال البحثي المتعلق برؤيا الحاسب وهي من المجالات البحثية النشطة. هناك العديد من التحديات التي تواجه الحاسب للتعرف على شكل جسم الانسان مثل الإضاءة، الملابس، البيئة الخارجية المحيطة بجسم الانسان.

على مدى السنوات الخمسة عشر الماضية، إكتسبت مشكلة التعرف على شكل جسم الانسان إهتمام العديد من الباحثين، كما تم طرح العديد من التقنيات من قبل الباحثين لمواجهة التحديات التي يواجها الحاسب للتعرف على شكل جسم الانسان. حتى الآن يتم إجراء معظم عمليات التعرف على شكل جسم الانسان على الأشخاص الذين يرتدون الملابس الغربية التي تقوم بكشف وإظهار جزء أو أجزاء من جسم الانسان. ومع ذلك فإن التعرف على أجزاء جسم الانسان في الملابس الغربية هو أسهل نسبياً من التعرف على أجزاء جسم الانسان لأشخاص يرتدون ملابس تعطي كل أجزاء الجسم مثل الثوب العربي.

وبناءاً على ذلك نهدف من خلال هذه الأطروحة التعرف على شكل جسم الانسان الذي يرتدي ملابس تغطي كامل أجزاء الجسم وبالأخص الأشخاص الذين يرتدون الثوب العربي، حيث يتميز الثوب العربي بأنه قطعة واحدة من القماش تغطي كامل جسم الانسان. وللوصول الى هدفنا المنشود في هذه الأطروحة تم إستخدام تقنية تعتمد على التعلم تستند الى البكسل (Pixel) والرقع (Patch) للتعرف على شكل جسم الإنسان وذلك من خلال تطبيق التقنيات المستخدمة في رؤيا الحاسب التي ساعدتنا على إكتشاف بعض المعلومات عن أجزاء جسم الانسان. وللحصول على معلومات كاملة أو توقعات حول كل جزء من أجزاء جسم الانسان تم إجراء التصنيف على هذه الميزات. كما أننا قمنا بإستخدام خاصيتين جديدتين للتعرف على شكل جسم الانسان في الصور.

حيث أظهرت النتائج قدرت النظام التي تم بناؤه خلال هذه الاطروحة للتعرف على شكل جسم الانسان الذي يرتدي الثوب العربي بنسبة 64% من خلال استخدام التقنية التي تعتمد على البكسل، بينما نتائج تقنية التعلم التي تعتمد على الرقع (Patch) كانت 65%.

CHAPTER 1

INTRODUCTION

From the last few decades, most of the computer vision problems such as object recognition and scene recognition were solved in parts. The image or a video are segmented into parts in order to recognize the objects. Human pose recognition is a kind of part based computer vision problem. The body parts are recognized from the whole body and using these recognized parts the exact human pose can be predicted. This problem gained the significant attention from the researcher from the beginning. Human pose recognition has an increasing number of new range application for example 3D gaming, sign language interaction, human-robot interaction, sports performance examination and human gait analysis. In spite of numerous research, human poses recognition is still a tough and unsolved problem. However, there exist some challenges in human pose recognition that were not completely encountered by the existing methods. The challenges include human physique, illumination, occlusions, human appearance, skeleton structure, human pose in an outdoor environment and the most important is human wearing a

western and draped dress.

Pose recognition in western clothes is comparatively easy as compared to draped clothes. There is a massive amount of literature available on western clothes pose recognition and with the announcement of Microsoft Kinect [10] this problem become easy and gained more focused from the researchers. The researchers start proposing new methods and algorithms to solve the pose recognition challenges using Kinect device. The Microsoft Kinect is a low-cost 3D motion sensor that can be used to develop the interactive application. This low-cost sensor solved many computer vision problems (e.g. pose estimation, human detection, 3D reconstruction etc.). However, all the methods and algorithms to date are specifically designed for western clothes. Shotton et al, 2013 [2] from Microsoft recognize real-time pose in parts using single Kinect depth image. The Shotton approach was rooted on object recognition strategies. They identify the human body parts and localize the 3D joint position. The machine learning classification techniques called Random Forest classifier was used for training and prediction. The Shotton approach was specifically designed and tested for western clothes like jeans, sports trousers, casual and dress shirts and it is invariant to the body, shape etc.

However, in draped clothes specifically Arabic thobe the Shotton [2] approach fail to predict human pose or even fail to detect human body parts as well [3]. As we know the Arabic thobe is covered with a single piece of fabric and it is very difficult to locate the body parts. Ridwan [3] in his MS thesis worked first time

on non-western clothes based human pose estimation specifically on Arabic thobe. He used almost the same features as used by Shotton [2] with same classification technique. He proposed new body part for lower body section called thobe body part. The results were not satisfactory for both upper body and lower body section the overall accuracy he got was 43%. The pixels for left thigh were incorrectly classified into the right thigh and same for the rest of the lower body section. These incorrectly classified pixels decline the accuracy for the lower body parts.

Therefore, in this research, we used Ridwan *et al.* [3] dataset and recognized human pose for people wearing draped clothes specifically for Arabic thobe. We developed a framework for recognizing human pose in draped based clothes. In this framework we adopted learning based technique in Pixel and Patch based methods for recognizing human pose. In each method we applied depth and computer vision feature extraction techniques. In pixel based method we applied HOG and SIFT as computer vision feature techniques. Whereas for depth features we used two new features called Median and LBP-decimal depth features. In Patched based methods we used Bag of features technique which is based on computer vision and local features as a depth. However, every feature we used in both methods has different characteristics. Therefore, to get benefit of that we decided to fuse these features together to further improve our results.

1.1 Background

Human pose recognition is considered a well-known process of estimating the human body pose from a single image or from video frames. It is one of the major problems in the field of computer vision. Due to its complexity, this problem has gained the focus of many researchers for over 15 years. Human pose plays an important role in the human communication process. The human posture is used to represents the different emotions. A recent study [11] shows that the human body poses gives better emotion than facial expression. Birdwhistlell [12] describe the human communication process. According to him, the words represent only 7% and non-verbal represent 55% of the communication process. Human pose recognition is a non-verbal communication process which is used to recognize the pose of a human. A pose can be eating, walking, sitting, discussion and waiting etc. Human pose can be recognized by localizing joints on human body and dividing the body joints into parts such as left head, right head, neck, left shoulder, right shoulder, left chest, right chest, left abs, right abs, left upper arm, right upper arm, left elbow, right elbow, left lower arm, right lower arm, left wrist, right wrist, left hand, right hand, left thigh, right thigh, left knee, right knee, left leg, right leg, left ankle, right ankle, left foot, and right foot. Using these segmented body parts, the human pose can be recognized accurately. There exist many applications that can advantage from human pose technology. For example, it is used in activity recognition and human-computer interaction(HCI) applications. It is also used in 3D gaming, character animation and clinical analysis of human gait.

1.1.1 Human Pose Recognition Methods

Pixel Based Method

Pixel-based methods are considered best for human pose accuracy. According to Ramanan *et al.* [13] pixel based methods can have better features if a pixel details used as input. This method can also be used collectively with other methods. For example, Shotton *et al.* [2] and Hernández-Vela *et al.* [14] used per-pixel wise classification method for recognizing human pose

Patched Based Method

Part based methods are different from pixel methods. In part base methods, the human body and parts are first spotted from an image then this method is applied to recognize the pose. This method is based on the position and appearance model. The method can be more effective if the exact body part position is known. Many studies [15] [16] [17] [18] [2] [19] [20] [21] [22] used this method for estimation of human pose.

Figure 1.1 shows the examples of pixel and patch based human pose recognition. In figure 1.1a pixel-based image shows that body part labels are depicted with different unique colors where as in figure 1.1b shows a sample image of different human body parts patches [23].

(a) Pixel-Based Image [24].

(b) Patch-Based Image[23].

Figure 1.1: Pixel and Patch based Methods

1.1.2 Human Pose Recognition in Western and Draped Clothes

In human pose recognition, most of the research has been taken on western clothes [2] since this problem first occurred. The structure of western clothes is very simple. The fabric is not covering the entire body. Some of the body parts are separated to each other for example left arm, right arm, left leg and right leg. That is why human wearing jeans, casual and dress shirts or sports trousers it is easy recognized upper and lower body parts using low-cost depth cameras. The low-cost depth camera like Microsoft Kinect or Asus Xtion sensor represents human body into Skeleton and provide joints data. Using depth sensor data provided by the Kinect, the human pose can be easily recognized on western clothes. However, there is no such research exists that recognized human pose on draped clothes. The draped clothes such as Arabic Thobe and subcontinent dresses are difficult to recognized human pose. These clothes are unlike western clothes it covers the entire body with a single piece of fabric. Therefore, in such cases the lower body would look like a concrete opaque square in a 2D image hiding all the spatial

details of lower body parts left thighs, right thighs, left knees, right knees, left leg and right leg. Even the low-cost depth cameras are failed to detect these body parts.

1.1.3 Kinect Depth Sensors

In 2010 Microsoft launched Kinect device that gained the attention of researchers to test their computer vision algorithms on low-cost sensing device. These depth cameras are widely used in computer vision research and have been used in many applications on the internet. 1.2 describes the internal structure of Kinect sensor. The sensor consists of RGB camera which is used to detect the color components (Red, Green, and Blue) and a depth sensor which uses both the IR emitter and sensor for depth computation for the scene. The RGB and Depth camera have 640 x 480 resolution with 30fps. Microsoft Kinect played important role in the recognition of human pose[25][26][27]. Moreover, researchers proposing new methods and algorithms to recognize full human body and all human body physical poses in order to evaluate the ability of Kinect since its first release. [28][29][30].

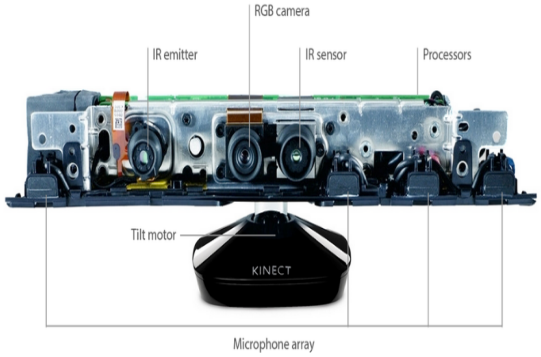


Figure 1.2: Structure of Microsoft Kinect Depth Sensor[4]

1.2 Problem Statement

There has been a lot of work done on human pose recognition from the beginning of this problem. All the research on human pose were specifically focused on western clothes like jeans, sports trousers, casual and dress shirts. However, until now nobody reported even a single published work on pose recognition in draped clothes except one MS thesis [3]. The draped based clothes are special kind of clothes which is very different to the western clothes. The whole human body or lower body section are covered with a single piece of fabric like Arabic thobe, subcontinent traditional dresses, skirt etc. Sometimes these fabrics have different deformation and hide the human body parts underneath. In these case, it is very hard to detect or recognize the human body parts. However, the Microsoft Kinect depth sensor also fails to give accurate skeleton information on draped clothes and Ridwan *et al.* [3] results are not very impressive for upper and lower body parts identification.

Therefore, in this thesis our primary target is to improve the results of Ridwan *et al.* [3] for both upper and lower body parts by applying computer vision and machine learning techniques.

1.3 Contributions

The contributions of this thesis are listed below.

- Developed the framework for the different draped based clothes.

- Used computer vision feature extraction techniques: HOG, SIFT and Bag of features.
- Introduced new depth features Median, LBP-Decimal and local depth features.
- Fused features together in order to check the dominance.

1.4 Methodology Overview

We followed learning based technique for pixel and patch based methods. figure 1.3 describe the general framework of our proposed work. Given depth and ground truth image as input. In part based method we randomly selected 800, 1000, 1200, 1400 and 2500 pixels from the depth image and apply different depth and computer vision feature extraction techniques on each randomly selected pixel. The selected features then classified using Random Forest machine learning technique to build a training model. The test data which consist of pixels from whole body is then passed to the training model in order to test the human body parts recognition accuracy. Whereas, in patch based method we first extracted the patch for each labeled body part in depth image. In each patch we applied Bag of feature technique which consist of HOG, SIFT and local eigen features. Finally, we fused all the features together to have a good comparison of different features.

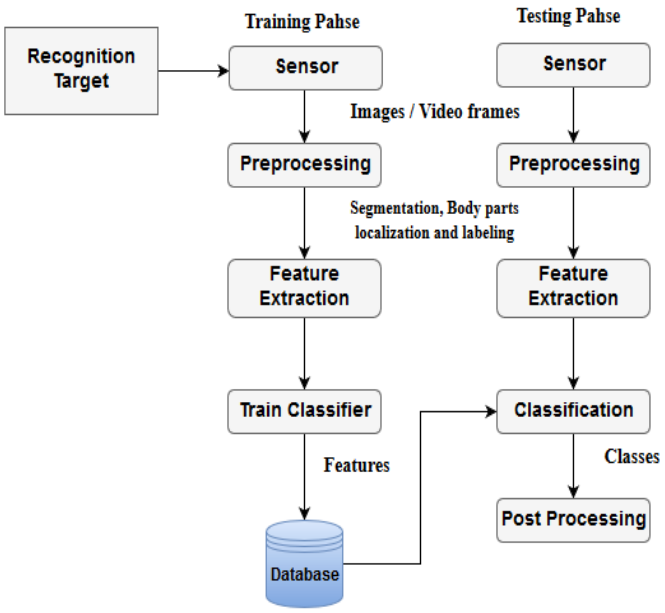


Figure 1.3: General Framework for Learning Based Human Pose Recognition[5]

1.5 Thesis Outline

The rest of the thesis is ordered as follows.

In Chapter 2 we presented related work on human pose recognition. Chapter 3 we described the methodology of human pose recognition in draped based clothes. In Chapter 4 we showed the experimental setup, parameters, datasets, experimental results and discussion about the results. Finally in chapter 5 we concluded our thesis work with limitation and the future direction of this thesis.

CHAPTER 2

RELATED WORK

2.1 Sensors for Human Pose Recognition

We conducted a little survey on time of flight cameras and depth sensors avaiable in the market. A list of other available sensors along with their classification and capabilities including 3D and RGB resolution, and frame rate. Table 2.1 shows the different parameters each sensor provides, which can be considered for recognizing human pose.

2.2 Preprocessing

2.2.1 Human Body Model

Selection of a human body model is one of the major factors in recognizing the human pose. The body model encloses information such as human texture and shape. In the literature, we found three types of human body models, namely,

Sensor	Type	3D Resolution	RGB Resolution	Frame Rate
Microsoft Kinect 2.0	Time of flight	512x424	1920x1080	30-fps
Asus Xtion Pro	Structured light	640x480	1280x1024	30-fps
Intel RealSense R200	Stereo and pattern projector	640x480	1920x1080	60-fps
IFM Efector	Time of flight	176x132	N/A	25-fps
Stereolabs ZED	Embedded stereo	2208x1242	2208x1242	15-fps
Carnegie Robotics	Embedded stereo	2048x1088	2048x1088	15-fps
Ensenso	Structured light	1280x1024	1280x1024	10-fps
SICK 3visitor -T	Time of flight	144x176	N/A	30-fps
e-Con System Tara Stereo	Embedded stereo	752x480	N/A	60-fps
Narian SPI	FPGA Stereo	640x480	N/A	30-fps

Table 2.1: Available Sensor for Recognizing Human Pose [1]

cylindrical human model, pictorial structure human model and kinematic human model. Human pose recognition is heavily dependent on these models whether it is used for full body pose recognition or specifically for recognition of upper body pose.

Cylindrical Human Model

Also called a volumetric model. It is used to represent both human pose and human body parts. In this model the human body parts are represented as fixed cylinders. Each cylinder consists of joints. For example, a single human arm represents three joints and these three joints represent one cylinder. The cylinder is further connected to other cylinders in order to form human body structure.

The meshes with cylindrical model can also be used to represent human body and its parts. Ganapathi *et al.* [22] represent the human body via meshes. Siddiqui *et al.* [31] proposed an approach that represents human body as a skeleton. The skeleton is then mapped with cylinders with fixed width. Ling *et al.* [32] proposed a similar cylindrical technique for tracking lower body parts such as the thigh, leg, calf, and foot.

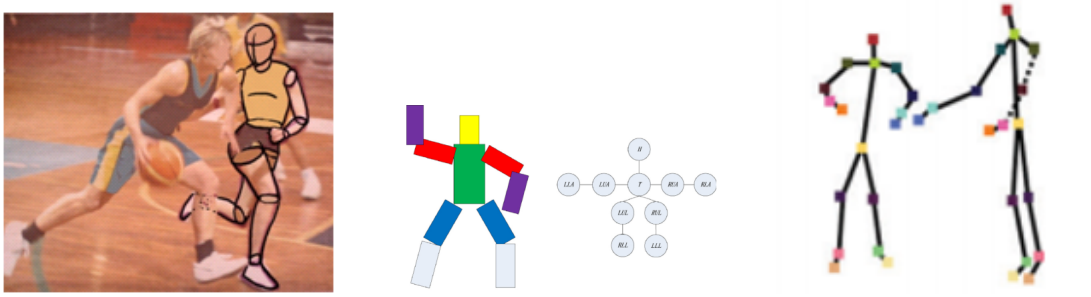
Pictorial Structure Human Model

This model is also a very famous model for recognizing human pose. The model represents human body parts as rectangular shape. Mykhaylo *et al.* [33] in 2009 used pictorial structure to predict human pose. Eichner *et al.* [34] [35] used the same model for human pose recognition.

Kinematic Human Model

With the announcement of depth sensor this model is frequently used nowadays. This model represents the human body as a set of joints. The human body model generated by the depth sensors consist of 30 to 32 joints depending on the depth sensor used. Using the 3D coordinates of human joints, the human pose is easily estimated. Shotton *et al.* [2] used Kinect sensor to compute human skeleton. Zequn *et al.* [15] used skeleton data for human pose recognition. Youness *et al.* [16] and Ishan *et al.* [17] also used the same model for human pose recognition.

Figure 2.1 shows the different models for recognition of human pose and human body part detection.



(a) Cylindrical Model [23] (b) Pictorial Model[36] (c) Kinematic Model[2]

Figure 2.1: Examples of different Human Body Model

2.2.2 Localization of Human Body, Joints, and Parts

Localization of human body, joints, and parts is one of the key steps in the pre-processing phase. Localization is the process of locating the position of human body, its parts and joints from a given image. Many authors [2], [15], [16], [17] use depth sensors like Microsoft Kinect to localize the human body and joints. The depth sensors provide the skeleton information of human. Using this information some techniques [15] [16] find the relative distance between joints and localize the human body parts. Localization of human body is also done through motion sensor devices. Marta *et al.* [37] used Vicon motion sensor to locate human joints.

2.2.3 Segmentation of Human Body Parts and Labeling

Ganapathi *et al.* [22] segmented the human body model into 15 rigid parts. Shotton *et al.* [2] divided the human body into 31 parts. These body parts are then labeled with unique colors in order to distinguish each body from another. Zequn *et al.* [15] divided the body parts into three regions, namely, body part, arm part and leg part Similarly, Youness *et al.* [16] also divided the whole body into 20

joints. Mingyuan *et al.* [24] divided the upper body section into 8 parts and labeled each part with a unique color. Ishan *et al.* [17] identified joints from Kinect and then with help of these joints the author segmented the body parts for pose estimation.

However, there are many public datasets available that use synthesized data and label the data with unique colors in order to represent different human body parts.

2.2.4 Background Separation

Background subtraction is another important step in the preprocessing phase. It is used to eliminate irrelevent details from the image by removing unwanted pixels. It was found that nearly every study found in literature [2], [38], [39], [40], [41] uses background separation.

2.3 Feature Extraction

2.3.1 Global Descriptor Based Feature Extraction

Histogram of Oriented Gradient (HOG) is a computer vision based global feature extraction technique. It is applied to the whole image and computes both vertical and horizontal gradients orientation and magnitude. This technique is normally used to detect humans in images. However, there is a vast amount of literature available that uses this technique for recognizing human pose.

In HOG, the image is divided into blocks, then the histogram of the gradient is computed for each block and finally, all histograms are concatenated to form a final feature vector. Figure 2.2 shows the HOG computation for a picture of a human. Sanzari *et al.* [37] estimate human 3D pose using Pyramid Histogram of Oriented Gradients (PHOG) visual features. They divide the human skeleton joints into groups and generate a dictionary of idiosyncratic motion snaps for each group. Each group contains the visual features while the groups are connected hierarchically. The purpose of a dictionary is to evaluate the probability of the group based on its visual features.

Wang *et al.* [42] used pose tree structure and applied HOG features on the human body. Similarly, Sun *et al.* [21] and Yang *et al.* [20] also applied HOG for features extraction. Eichner *et al.* [34] used pictorial structure-based model for recognizing human pose and used Edge HOG feature techniques on it. Fathi *et al.* [43] also used the same Edge HOG techniques to encounter feature vector using Hidden Markov Model. Fathi *et al.* [43] recognize human pose through video frames.

HOG is a global descriptor that is applied to the whole image instead of individual parts of the image. However, the human pose can only be predicted by first localizing the body parts. Therefore, this HOG technique may perform well for detecting human but may not give good accuracies for human poses. To solve this issue there is another technique called Deformable Part Multiscale Model (DPM) [44]. The technique is HOG based applied to individual body parts

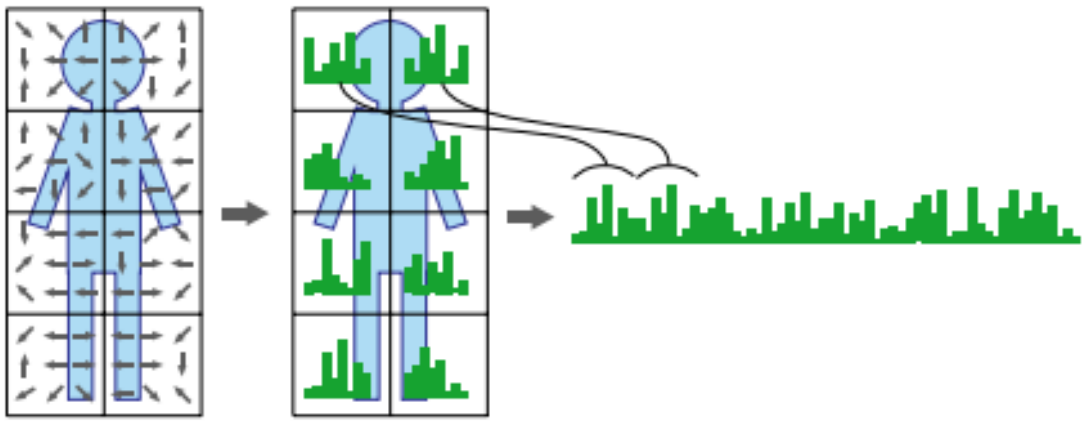


Figure 2.2: Histogram of Gradient (HOG) Computation [6]

instead of the whole image.

2.3.2 Local Descriptor Based Feature Extraction

A local descriptor like SIFT (Scale Invariant Feature Transform) and LBP (Local Binary Patterns) can be very effective for human pose recognition. These techniques are applied to each body part. The SIFT is gradient-based techniques. It calculates the orientation histogram for each cell and the resultant feature vector is the concatenation of all computed histograms.Figure 2.3 describe the SIFT computation. Ganapathi *et al.* [22] used local descriptor and Holt *et al.* [19] used local binary features for recognizing human pose.

2.3.3 Skeleton-based Feature Extraction

Skeleton-based features are normally calculated from depth sensors like the Microsoft Kinect sensor. The depth sensor gives the 3D coordinates of human joints. Using the relative distance between the joints, the feature vector can be computed.

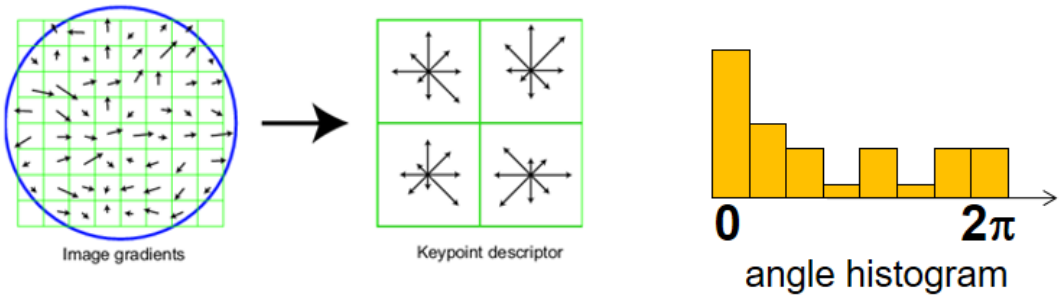


Figure 2.3: SIFT Computation

Youness *et al.* [16] calculate the set of 20 features from each pose using Microsoft Kinect skeleton data. The features are invariant with respect to position and size. Figure 2.4 describe the chosen features from Kinect skeleton information using relative distance. Similarly, Zhang *et al.* [15] identify 9 features from Kinect depth sensor. The authors calculate the features by computing the relative distances between joint pairs. The features include left forearm, right forearm, left upper arm, right upper arm, left thigh, right thigh, left crus, right crus and finally the spine. In 2015 Ishan *et al.* [17] also calculates the feature with the help of the Kinect sensor. The authors acquire skeleton information from the sensor and use velocity, position, and acceleration in feature computation. Siddiqui *et al.* [31] also use skeleton joints as features.

2.3.4 Depth Based Feature Extraction

Depth feature is calculated from depth images produced by the depth sensors. Shotton *et al.* [2] use depth features for recognizing human pose using random forest classifier. Contrary to analyzing the color data of an acquired image, the method extracts features by analyzing the depth information collected in the

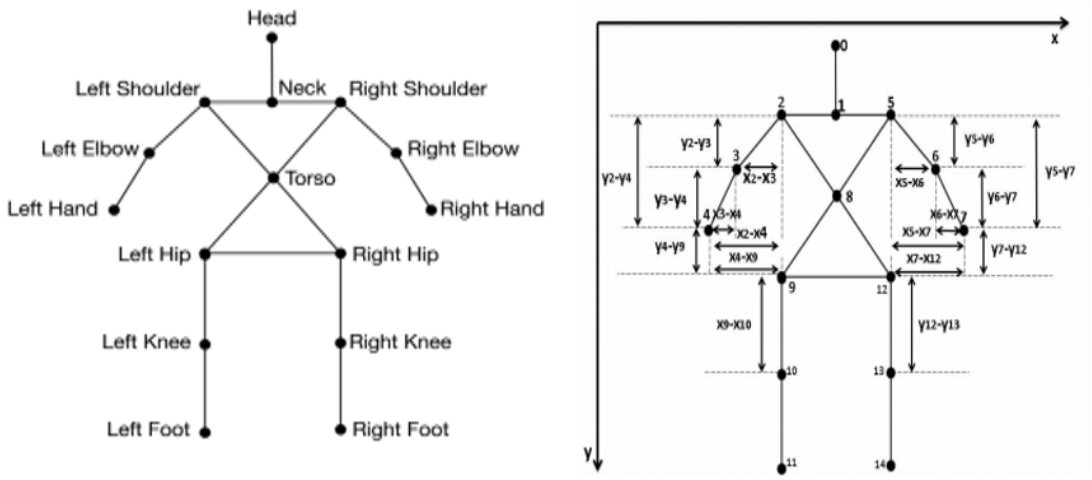


Figure 2.4: Chosen Features from Skeleton

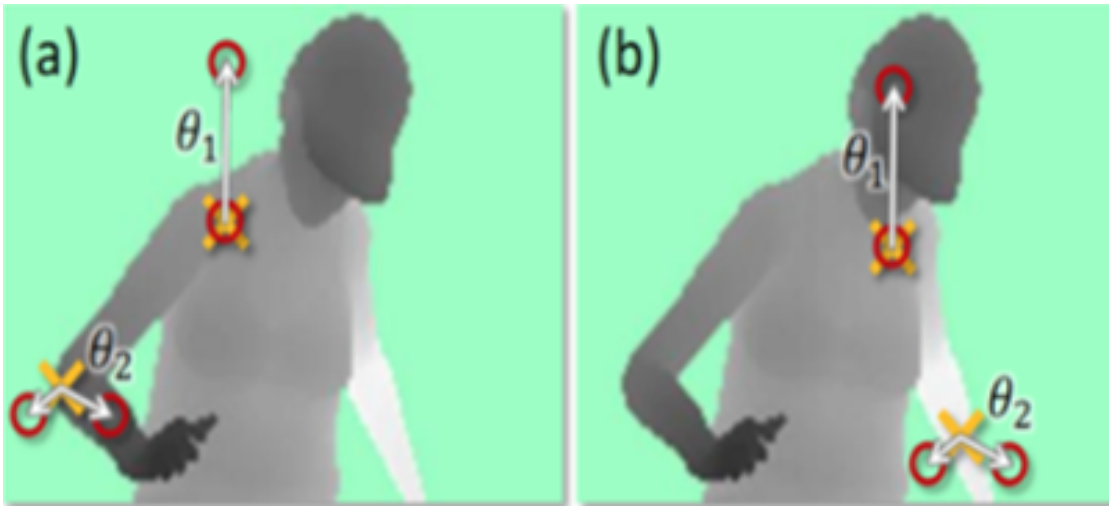


Figure 2.5: Shotton [2] Depth Feature Representation

depth image by the sensors. Figure 3.2 shows the Shotton *et al.* [2] feature representation. In figure a and b the two red circles indicate offsets and a yellow cross indicates the classified depth feature by taking the difference of two offsets. If the offset pixel lies outside the image or lies in the background, then the depth of the feature would have large positive constant value. Otherwise, the depth would have smaller response value.

2.4 Classification

Machine learning techniques are used to train classifiers for human pose recognition. The classifier is first trained with training feature dataset then test feature set are used with the trained classifier for prediction. Table 2.2 summarizes the work done with each classifier. Furthermore, studies exist in literature [33], [16], [43] that used AdaBoost, K-nearest neighbor classifier, and Hidden Markov Model respectively. Table 5 presents a comparison of classifier accuracies in human pose recognition.

Classifier	**Relatedwork**
Random Forest (RF)	[17][2][19][42]
Support Vector Machine (SVM)	[35][15][16][32][21][45]
Bayesian and Naive Bayesian (NB)	[37][22][31][34][16][17]
Artificial Neural Network and Deep Learning	[18][24]

Table 2.2: Classifier Related work

2.5 Post Processing

After successful classification of features, the post-processing phase defines the classes for the human pose. A pose can be classified as eating, walking, sitting, discussion and waiting to mention a few. Each pose mentioned belongs to a separate class.

Youness *et al.* [16] recorded total 18 poses and each pose consists of 20 features. Marta *et al.* [37] recorded 15 poses. Ishan *et al.* [17] define their own poses in order to detect emotions of designer team member. They used, engage, frustration,

boredom and neutral as poses. Similarly, Zequn *et al.* [15] recorded a total 22 different human poses. They divide these 22 pose into 3 categories. The categories are body part, arm part, and leg part. The body part category has 7 different poses, while the arm part has 8 different poses and finally the leg part has 7 poses. Some of the authors recognized upper and lower body pose based on dataset they used.

2.6 Datasets

There are many datasets available for human pose recognition. Some of the datasets are specific to upper body parts, others are specific to lower body parts and yet others cover the whole human body parts. Table 4.1 lists down the available dataset for human pose.

Datasets	**Contents**	**Type**
HumanEva [46]	50,600 training frames, 26,400 testing frames	Full
EVAL [47]	24 sequences	Full
LSP [48]	1000 training images, 205 testing images	Full
Parse [49]	100 training images, 276 testing images	Full
SMMC-10 [22]	6 performers, 28 sequences	Full
PDT [50]	26,400 testing frames, 40 sequences	Full
FLIC [51]	3987 training images, 1016 testing images	Full
PASCAL 12 [52]	Total 20 classes, 11,530 images for training and validation	Full
Buffy [53]	748 frames from "Buffy the vampire slayer" TV show	Upper
MPII [54]	410 activities of different color and sizes	Full
Poses in the wild [55]	30 sequences of different color and sizes	Upper
Human 3.6M [56]	3.6 million images with 17 scenarios	Full
CMU [57]	23 actions, 109 subjects and 2605 videos	Full
MPII Cooking Activities [58]	65 actions, 12 subjects, and 44 videos	Upper
UMPM [59]	Multiple people with 30 subjects and 36 videos	Full
TUM Kitchen [60]	4 actions, 4 subjects, and 20 videos	Full
KTH Multiview Football [61]	Total 8307 images with 2D and 3D dataset	Full
Video Pose [62]	Total 1289 images from 44 short clips	Upper

Table 2.3: The Available Datasets

Methods	**Features**	**Classifiers**	**Datasets**	**Acc**
Marta [37]	PHOG	Hierarchical Bayesian	Human 3.6 M	-NA-
Youness [16]	20 features using relative distance	SVM, ANN, KNN and Naïve Bayes	Real data from Kinect	Fig 2
Ishan [17]	Velocity, acceleration and position	C4.5, Random Forest, IBK, Naïve Bayes	Real time frames from Kinect	98%
Zequn [15]	9 features using relative distance	SVM	Real time frames from Kinect	99.14%
Toshev [18]	-NA-	Deep Neural Networks	LSP, FLCIC and Image Parse	69%
Jiu [24]	Energy function	Deep Learning	CDC4CV Poselets dataset	66.92%
Wang [42]	HOG	Tree style	PARSE, LSP	62.8%
Shotton [2]	Depth features	Random Forest	Own created dataset	60.30%
Eichner [35]	HOG, Shapes Edges	SVM	PASCAL 08, Buffy	-NA-
Sapp [45]	Geometry, Color optical	SVM	Video Pose 2.0	68.3%
Sun [21]	HOG	SVM	PASCAL 07	64.2%
Yang [20]	HOG	SVM	LSP, Image Parse and Buffy	55.1%
Holt [19]	Binary features	Random Forest	CDC4CV Poselets	67%
Ganapathi [22]	Local descriptors	Dynamic Bayesian Model	MOCAP from Phase Space System	-NA-
Siddiqui [31]	Skeleton joints	Bayesian	Real time frame from SR300 sensor by MESA	0.930
Eichner [34]	Edgelet HOG	Probability based	Own dataset	-NA-
Mykhaylo [33]	Shape Context	AdaBoost	TUD-Pedestrians, TUD-Upright People	55.2%
Fathi [43]	Edges HOG	HMM	CMU MoBo	-NA-

Table 2.4: Related Work Summary

CHAPTER 3

THESIS METHODOLOGY

3.1 Pixel-Based Approach

Pixel-based method is considered best for human pose accuracy. However, by using this method the background can be easily separated from foreground object based on the pixel information. Therefore, by getting the advantage of background separation we used this method in our thesis. We used Ridwan *et al.* [3] dataset. In his dataset the background pixels have already assigned a fixed constant value.

In Thobe dataset we have two types of images. These two images consists of depth and ground-truth image. Both image have same size and resolution. The depth images contains the depth value which basically represents a distance from depth sensor. The range of depth value is between 0 to 4000mm. the background has constant value which is 50000. Where as the ground-truth image is consider as labeled RGB image which give us an information about pixel that from which body part the pixel is belong to.

Given a depth image we first randomly collected different number of pixels from whole image. The different number of pixels are 800 pixels,1000 pixels,1200 pixels,1400 pixels and 2500 pixels. The reason of selecting these random numbers of pixels over whole body is following. First, to compute whole body pixels for training dataset is computationally expensive. Because the training dataset is too large to evaluate every pixels. Second we analyzed the different random numbers of pixels and as per our analysis these different random number of pixels covered almost every body part in the whole image. We showed this analysis in chapter 4.

We applied both depth and computer vision feature extraction techniques on every randomly extracted pixel. Lastly, we classified every feature extraction technique into random forest classifier. We also classified the combination of two or more feature extraction techniques together. Figure 3.1 shows our pixel based framework.

3.1.1 Depth-Based Features Extraction

Median Depth Feature(MDF)

Shotton *et al.* [2] used to calculate simple depth feature by using 3 random pixels. In order to classify random pixel x. Given a depth image shotton *et al.* [2] collects two more random pixel(u, v) from depth image called offset pixels. The offset distance between the classified pixel and offset pixels is 198. The offset distance means that the two random pixels taken in the radius of 198 pixels around

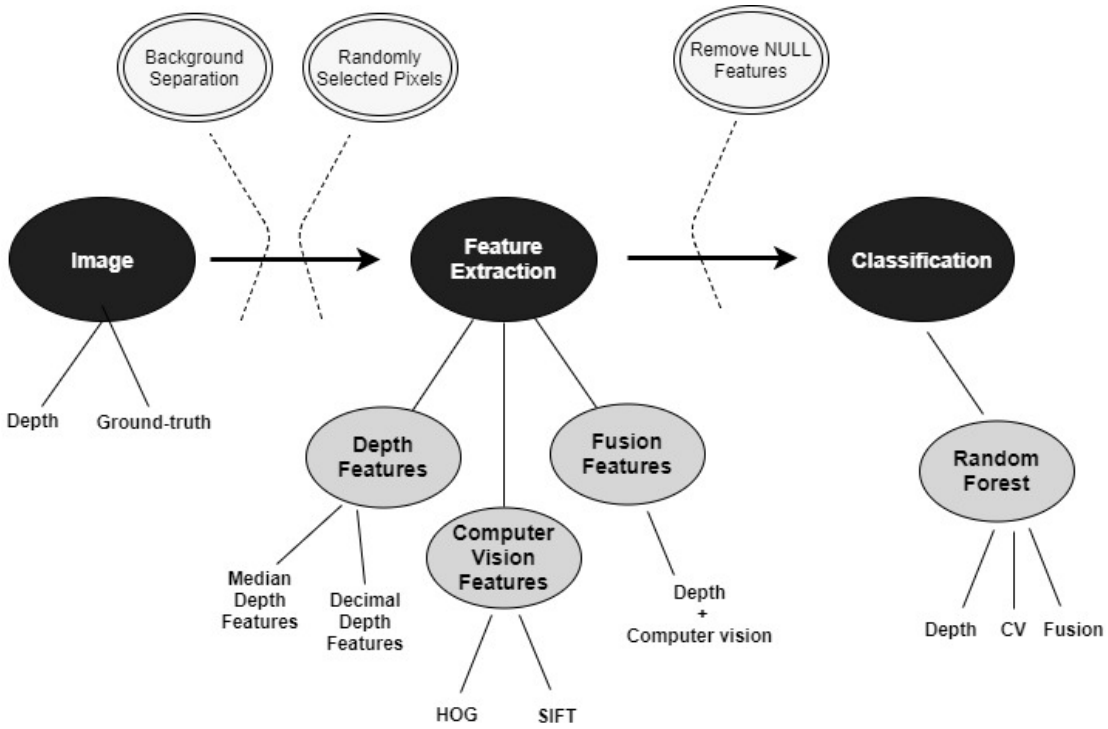


Figure 3.1: Framework for Pixel based technique

classified pixel x. The depth feature is calculated by following equation.

$$f(I, X) = d(X + \frac{u}{dx}) - d(X + \frac{v}{dx}) \tag{3.1}$$

In equation 3.6 the d is the depth, dx represent depth of pixel x , (u, v) are two offsets and X is the coordinates of classified pixel $x1$.

Figure 3.2 shows the Shotton *et al.* [2] feature representation. In figure a and b the two red circles indicate offsets and a yellow cross indicates the classified depth feature by taking the difference of two offsets. If the offset pixel lies outside the image or lies in the background, then the depth of the feature would have large positive constant value. Otherwise, the depth would have smaller response value. Due to high computational complexity Shotton *et al.* [2] used two offset

pixels (u, v) instead of region average.

Similarly, by the follwing the technique of Waldvogel [63] and Shotton *et al.* [2]. Ridwan *et al.* [3] implemented region average in his MS thesis. To classify pixel x Ridwan *et al.* [3] take two random regions$(R1, R2)$ in a depth image. In Figure 3.3, the two offsets u, v, depicted as white arrows, the two widths $(w1, w2)$ and the two heights $(H1, H2)$ are selected at random. $R1$ and $R2$ are two regions centered around u, v with dimensions $(w1\text{x}H1)$ and $(w2\text{x}H2)$ respectively. The response to the query pixel depicted as a yellow cross is calculated as the difference between the average depth of region1 and region2. The depth feature is then normalized by dividing the difference of averages by classified pixel x.

However, by following the region averages feature technique Ridwan *et al.* [3] did not care about the noise and outliers values. The outlier values can make a big difference in feature response. For example in figure 3.3 region $R1$ at middle of the figure have few background pixels. The average of region $R1$ is become high because background pixel has high intensity value.

Therefore, we encounter this flaw and introduce new feature called median depth feature. The median depth feature take care of outlier values in the regions. We randomly take the two regions in the depth image and take the median value of that regions. Initially, we fixed the region size to 3x3 window, 5x5 window and 7x7 window. We calculated median depth features for all these windows separately. However, the values of depth image is different than other type of image. In depth image two or more parts may have same depth values because

of having same distance from the depth sensor. Therefore, to differentiate same depth the fixed region/window size may not work properly. Because there is a probability that two or more classified pixel x may have spotted the same region with same window size.

We solved this problem by taking the region height and width randomly. In this case if two or more classified pixel x may spotted same region but have more probability that the region height and width is different. This help us to differentiate the same depths. The two region may different or same width and size. The feature response of median depth feature is as follow.

$$f(I, X) = \frac{median(R1) - median(R2)}{d(X)} \tag{3.2}$$

In equation 3.2 the median depth feature have following parameters:

- I is a depth image.
- X is a classified pixel point.
- R1 and R2 are random region.
- d(X) is a depth of pixel x and it is used in equation 3.2 for normalization.

Figure3.4 shows the steps for calculating the median depth feature. The yellow cross in figure 3.4a indicate the random classified pixel x. To calculate the feature at pixel x we randomly spotted two region $R1$ and $R2$ in figure 3.4b. Finally, figure3.4c shows the median of two region using these two median we calculated the feature of pixel x by taking the difference of these two medians.

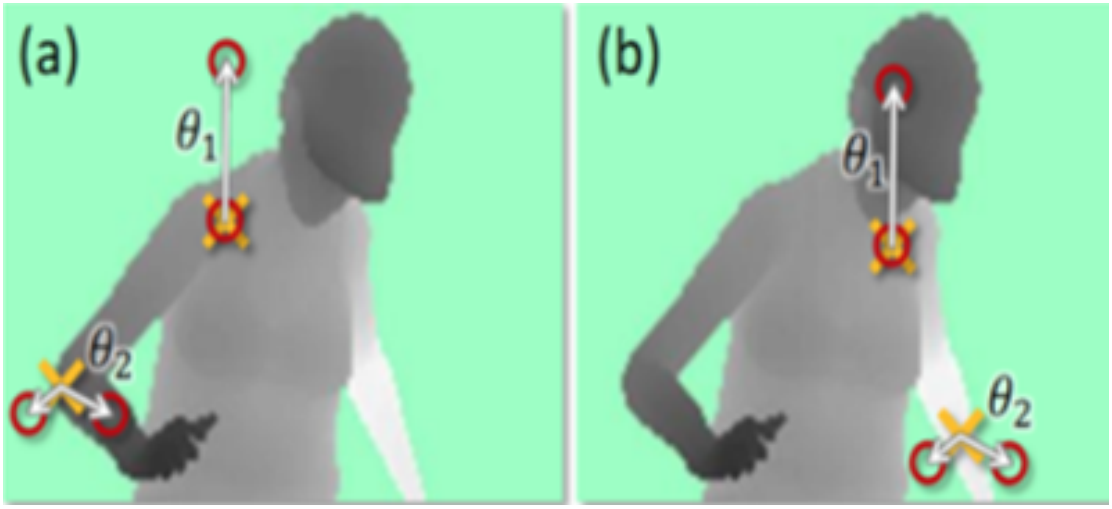


Figure 3.2: Shotton [2] Depth Feature Representation

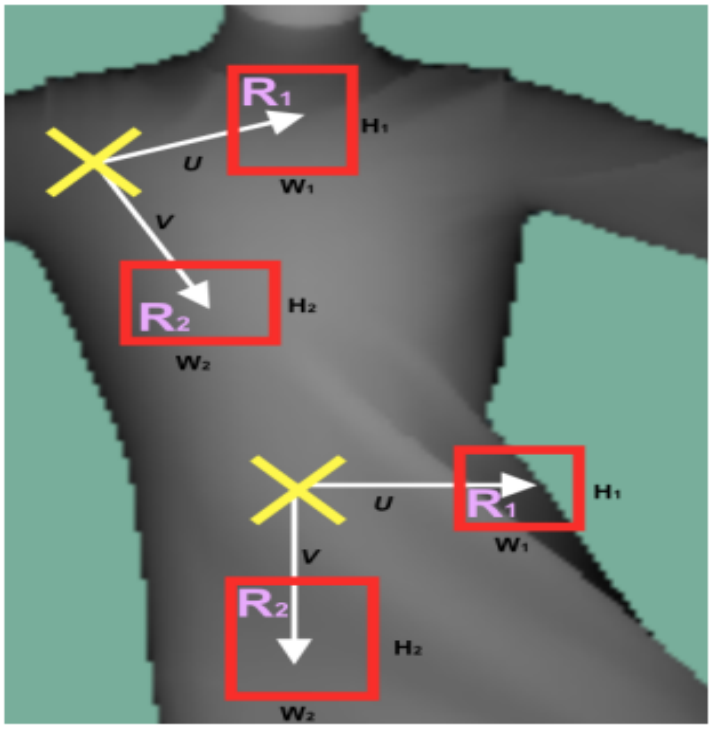
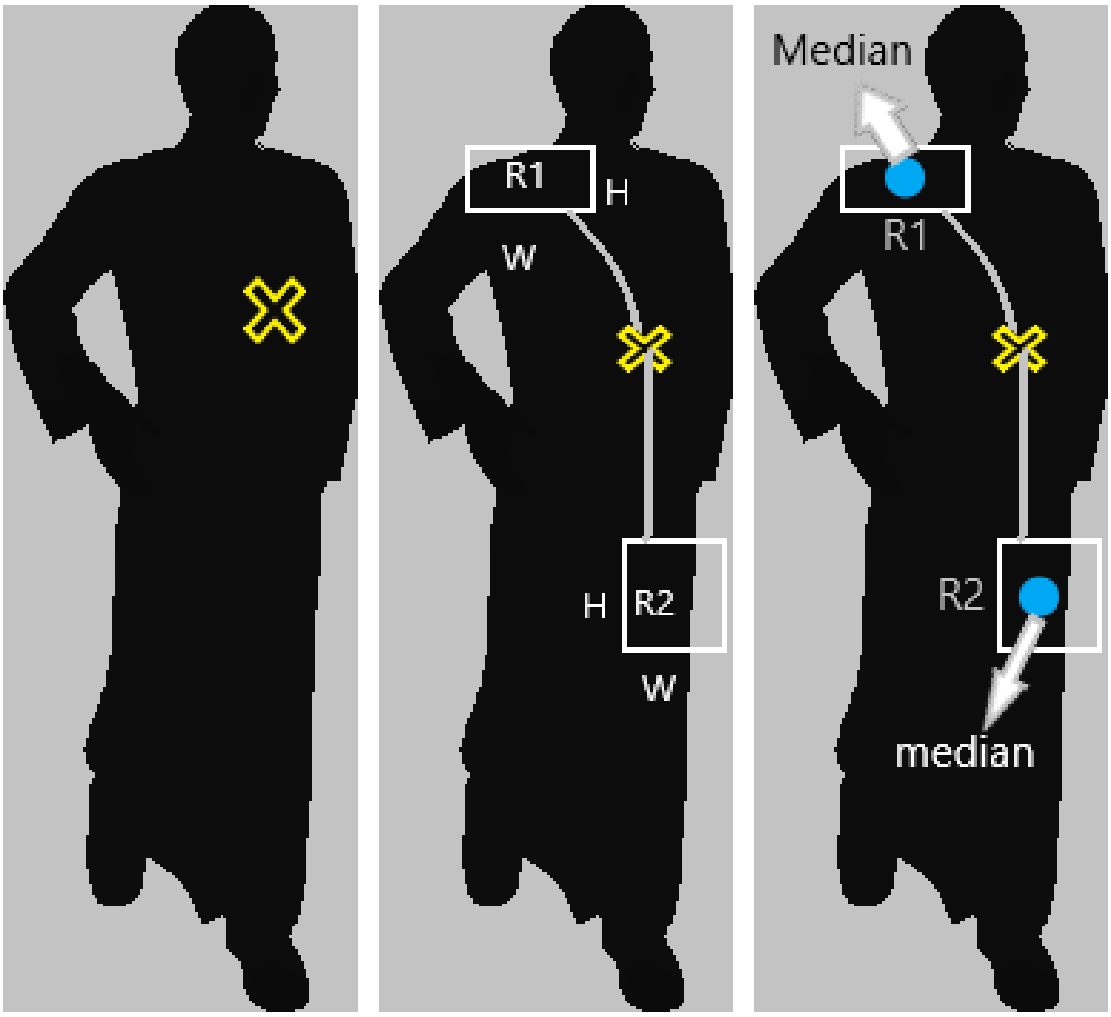


Figure 3.3: Ridwan [3] Depth Feature Representation



(a) Classified pixel x (b) Region R1 and R2 (c) Median depth feature

Figure 3.4: Median depth features representation

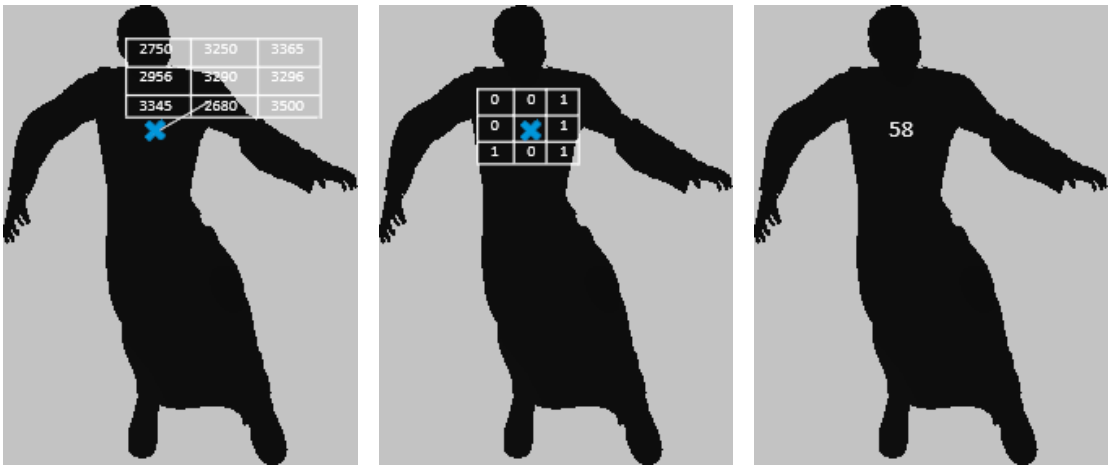
Local Binary Pattern-Decimal Depth Feature(DDF)

Inspired from the work of Ojala *et al.* [64] we introduce a new depth feature called local binary pattern decimal depth feature. This feature never used in depth image before. From now we call this technique decimal depth feature. This technique has different characteristics. Instead of taking averages or performing arithmetic operation on region or an individual pixel. We fixed a 3x3 window at a random position in a depth image. The centre value of 3x3 window is compared with its neighbour. If the neighbour pixel value is less than equal to center value we assigned that pixel value to 0 otherwise 1. By doing this we got size of 8 binary vector and then we converted this binary vector into a decimal number. The decimal number is representing a depth feature of the center pixel x of 3x3 window. In order to differentiate same depth in decimal depth we concatenate the pixel location as feature with decimal depth feature. Further details about feature vector is provided in section 3.4. We applied the same procedure for other different random pixels for a depth image. The feature response of decimal depth feature is as follow.

$$f(I, X) = \sum_{i=1}^{K} g(xi, xc)2^{k-i} \tag{3.3}$$

$$g(xn, xc) = \begin{cases} 1, & \text{if } d(xc) \leqslant d(xn) \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

Given an depth image I and random pixel at point X. We construct a 3x3 window around point X. The pixel xi correspond to the neighbour pixel where as



(a) 3x3 window at X (b) Binary representation (c) Decimal depth feature

Figure 3.5: Decimal depth features representation

xc correspond to the centre pixel of 3x3 window. we used $K = 8$ because of 3x3 window. Figure 3.5 shows the steps for calculating the decimal depth feature.

3.1.2 Computer Vision-Based Features Extraction

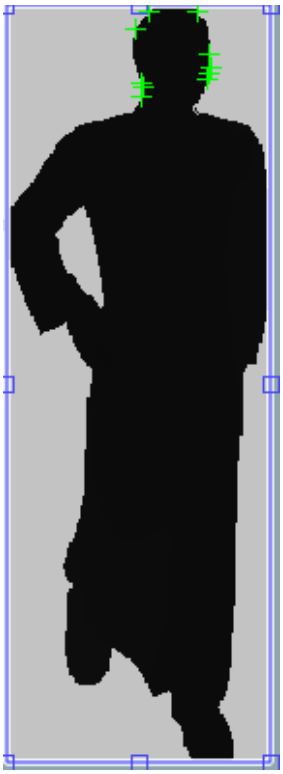
Feature Detectors

Feature detectors are used to identify the features in the image. These detected features are also called valid key points. We applied computer vision detectors on our region of interest. It decide at every image point whether it is a valid feature or not and provide us abstract information about region of interest. These detected features is a subset of image domain.

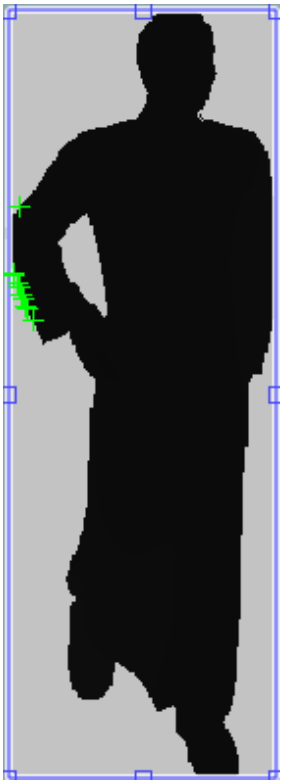
Before applying feature descriptor on every pixel. We tried key points detectors first on the depth image in order to find some valid key features from depth image and then feed those features or key points in to the feature descriptor. We applied following detectors:

- FAST Feature Detector.
- HARIS Feature Detector.
- SURF Feature Detector.
- BRISK Feature Detector.
- MSER Feature Detector.

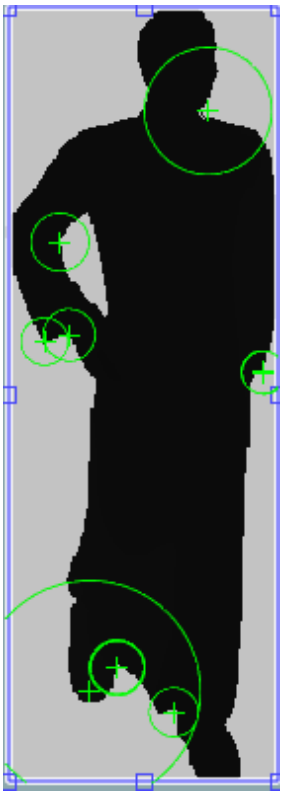
Most of the detectors are used to detect the corner points in the image. Using depth image on these detectors we did not able to detect the valid key points. However, the depth image is consist of depth values and doesn't have clear edges because of the nature of Arabic Thobe dress. Therefore, these detectors are meaning less for us specifically for the depth image. We used these detectors on ground truth image and translate valid points into depth image but this technique did not work for us as well. Therefore, we used feature descriptor on random pixels in a depth image. Figure 3.6 shows the results the applied detectors on a depth image. In figure 3.6a we only able to identified two body parts head and face key points where as the FAST detectors unable to detect rest of the features in remaining 25 body parts. Similarly, in figure 3.6b only able to detect left elbow features. These detected features are representing Thobe dress instead of representing the body part. The rest of the detectors are failed to detect all body parts features.



(a) FAST detectors

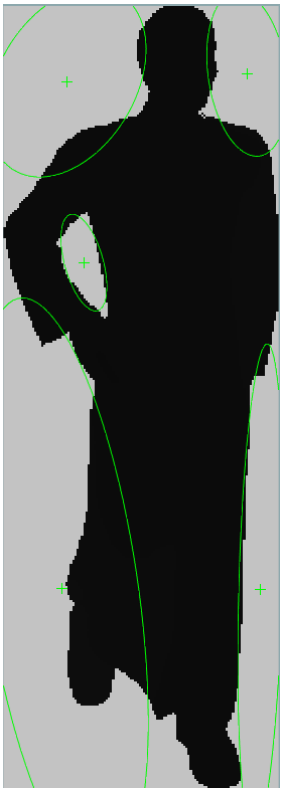


(b) HARIS detectors



(c) BRISK detectors

(d) SURF detectors



(e) MSER detectors

Figure 3.6: Results of feature detectors in a depth image

Histogram of Oriented Gradient (HOG)

After getting no valid key points information from detectors. We decided to use feature descriptor on randomly selected pixels in a depth image and feed that features to the classifier. A Feature descriptor is very helpful for recognition and detection of one or more object from an image. It is used to extract a meaningful information from an image or patch and ignores the extraneous information. These feature vectors are fed into classification algorithms such as Random Forest in order to produce good recognition and detection results.

HOG is one of very well known feature descriptor introduced by Dalal and Triggs [65] back in 2005. This descriptor works very well on human and pedestrian detection. The HOG feature vector is calculated by help of oriented gradient directions. These gradients occurrences are distributed using histogram which then represent as feature vector. The whole image is divided into number of different cells. The cell size can by 8x8, 16x16 and 32x32. In each cell the vertical and horizontal gradient directions is calculated on pixels. This is easily done by filtering the image using 1D kernel. All the vertical and horizontal gradients directions are concatenated together and represent it on histograms. The other implementation steps of HOG descriptor are as follow:

- To represent gradients into angular bins the gradients in each cell converted to some discrete values.
- Each pixel within a cell has some weighted gradient that correspond in angular bin. The angular bins corresponding to angles from 0 to 160.

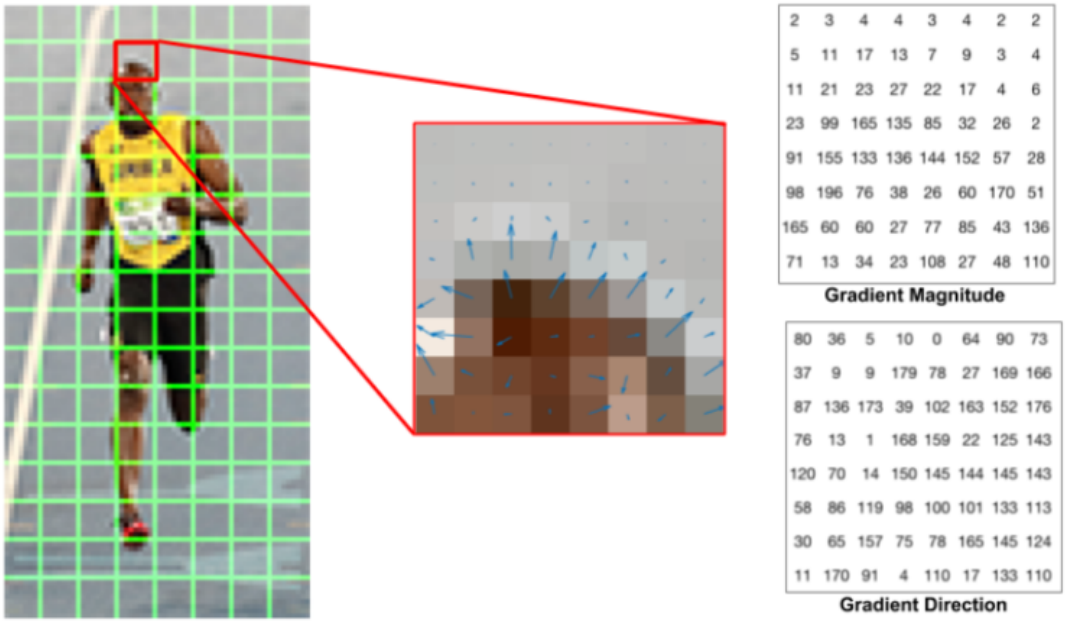


Figure 3.7: HOG descriptor

- The set of adjacent cell are considered as spatial connected blocks. These blocks are then normalized in order to avoid the lightening variations.

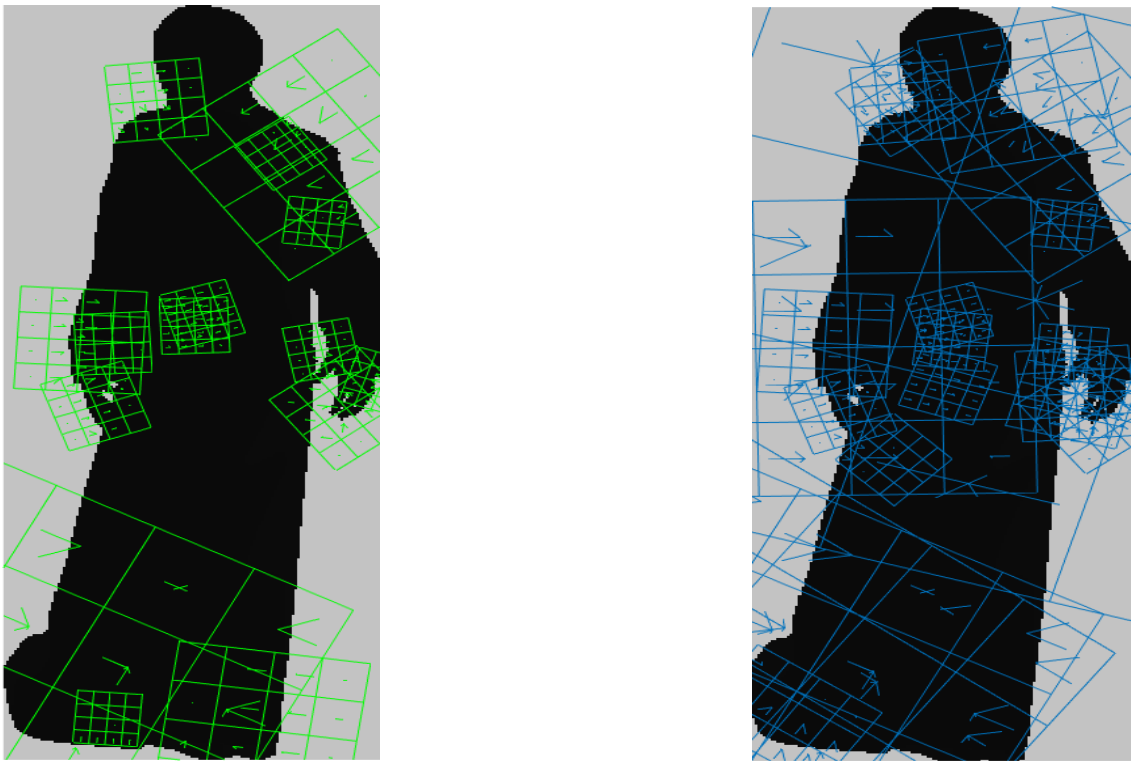
The set of blocks are represented as feature descriptor for HOG. Figure 3.7 and figure 2.2 shows the implementation of HOG descriptor where the image is divided into cells. The gradients for each is represented using blue arrows. The gradient magnitude of a cell represent a discrete numbers for a gradient direction. Figure 3.8 shows the HOG visualization in Arabic dress where the dominant gradient directions capture the shape of person in a Arabic dress.

We used the same configurations as proposed in [65]. According to Dalal and Triggs [65] these configurations produced better results in human detection and pedestrian detection . Usually HOG is applied on the whole image or on patch. However, we applied HOG at randomly selected pixels in a depth image. We

Figure 3.8: HOG visualization

extracted different number of random pixels from depth image such as 800, 1000, 1200, 1400 and 2500 random pixels. We used HOG features in two ways. First, in each random point we constructed a window around it. The window calculate the oriented gradients and concatenate all gradients into one histogram. The resultant histogram is the feature vector for a classified pixel x at point X. A second way is to use the same window that we constructed using first way and take the average of it. We then sum up the average value with the classified pixel x depth value and considered it a new depth feature in order to improve the human part detection performance using HOG and depth features together.

Sometimes, we get a random pixel on the border of the image where the constructed window lies outside the image bounds so in that case HOG return



(a) Depth HOG descriptor (b) Depth SIFT descriptor

Figure 3.9: Features descriptors on depth image

NULL feature vector. In this scenario we discard NULL values and consider only valid HOG features values. Figure 3.9a shows the HOG representation on a depth image.

Scale Invariant Feature Transform (SIFT)

SIFT is a computer vision algorithm which is used to detect object in an image. For every key point the SIFT descriptor provides a set of features that describe the region of interest. This algorithm introduced by Lowe *et al.* [66] in 1999. The algorithm is very useful for human detection and object recognition. It transformed the image into local features that is invariant to rotation, scale, translation and works very well under illumination variations. SIFT is similar to HOG except

that HOG describe global features where as SIFT describe local features in the image.

The basic principle of SIFT algorithm is smoothing and resizing an image into different scales like pyramids. The difference-of-Gaussian (DOG) [67] [68] function is applied in 3D pixel coordinates of an image by using local extrema. The valid local extrema are the keypoints. The local extrema are used to elimnate the noise from an image and gives better accuracy. The SIFT algorithm consist of following steps.

- The difference-of Gaussian pyramid(DOG) is created from an input image. This is done by convolution of the input image with different scales of Gaussian function(kernels). The DOG is calculated by taking the difference of scaled images

- Estimate the scale and location of keypoints from an image using Taylor series expansion. This step is called extrema detection.

- Once the local extrema are found the refinements is performed on keypoints in order to improve the accuracy of keypoints locations. The poorly localized keypoints are eliminated from the candidate keypoints.

- Estimate the orientation of every candidate keypoint. This is done by creating a histogram from the oriented gradients of keypoints within a region. Figure 3.10 shows the gradient orientation assignments. The black dot on the left side of the figure 3.10 is the candidate point. The black arrows

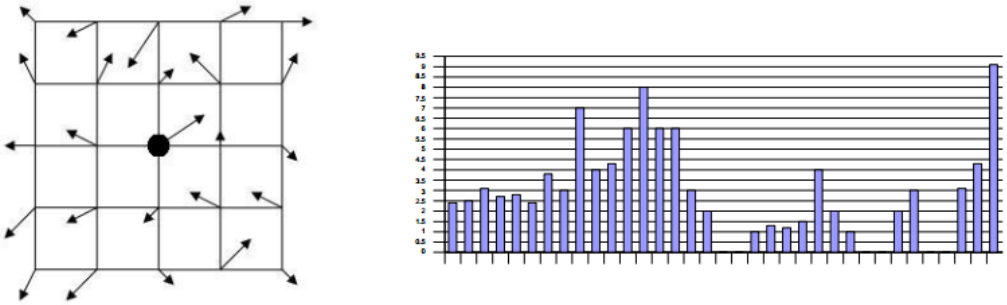


Figure 3.10: Estimation of gradients orientations [7]

around candidate point are the gradient orientation which is calculated using pixel differences around candidate point. The right side of figure 3.10 represent the 36 bins histogram. The value of each bin contains the sum of gradient magnitude for all all orientation within in a region/window.

- Estimate the features by applying the SIFT descriptor on every keypoints within a region. Figure 3.11 shows the SIFT keypoints descriptor. The left side of the figure 3.11 represent the gradients orientation and magnitude within 8x8 region/window with 4x4 sub-regions. The orientation of each sub-region is concatenated into one as we can see on the right side of the figure 3.11. The 8x8 window is concatenated into 2x2 block. Each block contains 8 direction which is represented by arrows. The computed 2x2 block orientations and magnitudes are then mapped into histogram that represent the feature vector for candidate point which is at the left side of figure 3.10.

However, the SIFT keypoints detector wasn't able to perform well on our

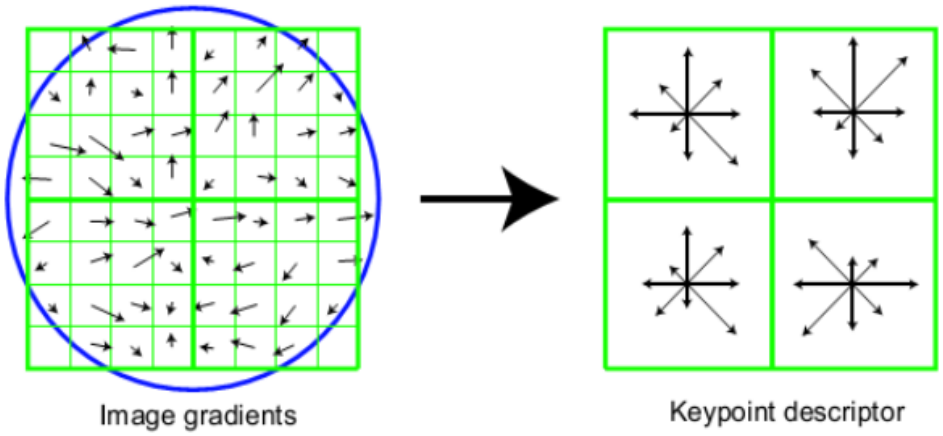


Figure 3.11: SIFT keypoints descriptor [8]

depth images. Like HOG we also applied SIFT descriptor at randomly selected pixel points in a depth image. We used different number of random pixels from depth image such as 800, 1000, 1200, 1400 and 2500 random pixels. In each point the descriptor constructed a window around center pixel x at point X. When we construct a 8x8 window at random point it become a patch which has same size as window size.

Sometimes we get a random pixel on the border of the image where the constructed window lies outside the image bounds so in that case SIFT return NULL feature vector. In this scenario we discard NULL values and only consider only valid SIFT features values. Figure 3.9b shows the SIFT representation on a depth image.

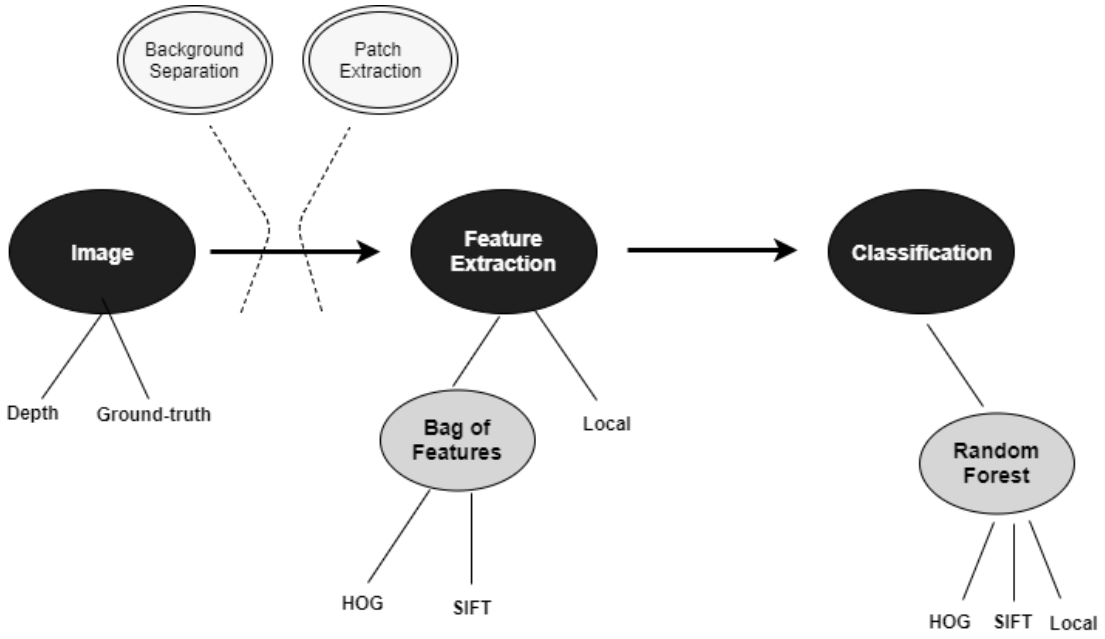


Figure 3.12: Framework for Patch based technique

3.2 Patch-Based Approach

Patch or part based methods are different from pixel based methods. In part base method, the human body and parts are first spotted from an image then this method is applied to recognize the pose. This method is based on the position and appearance model. The method can be more effective if the exact body part position is known. Figure 3.12 shows our patch based framework.

3.2.1 Bag of Features

In last few years a new computer vision technique called Bag of Features [69] have seen rising and it is used in many applications. The technique has been used in for object detection, image classification, reboots and image retrieval. It uses order less collection of image features and The image representation is analogous.

In terms of performance this technique is powerful enough that it outclass other state of the art methods in many applications.

In this technique a feature vocabulary is constructed. In order to create feature vocabulary a discrete clustering is applied on the set of training images. Clusters are needed so that the discrete local features vocabulary is generated from the large number of training samples. Whereas, in testing given an image features are extracted and assigned to the nearest cluster in discrete vocabulary. Afterwords, a normalized histogram is generated from the testing set that represent the actual features.

Due to its simplicity and performance we adopted this technique in our problem domain. Instead of images we extracted the body parts from the image as patch. We applied HOG and SIFT feature descriptor separately on each patch. Once we have a patch features from the all the training set we clustered the data in order to create a discrete vocabulary. We used K-Nearest Neighbour (KNN) clustering technique. We cluster the training data up to 200 clusters.

Given a test image the patches are extracted from the image. The computer vision techniques HOG and SIFT are applied on each patch respectively. The each feature which is extracted from the patch is assigned to the nearest neighbour in discrete vocabulary using minimum euclidean distance. At the end we would have a normalized histogram that actually represent the features of each patch. Figure 3.13 shows the steps involved in bag of feature technique.

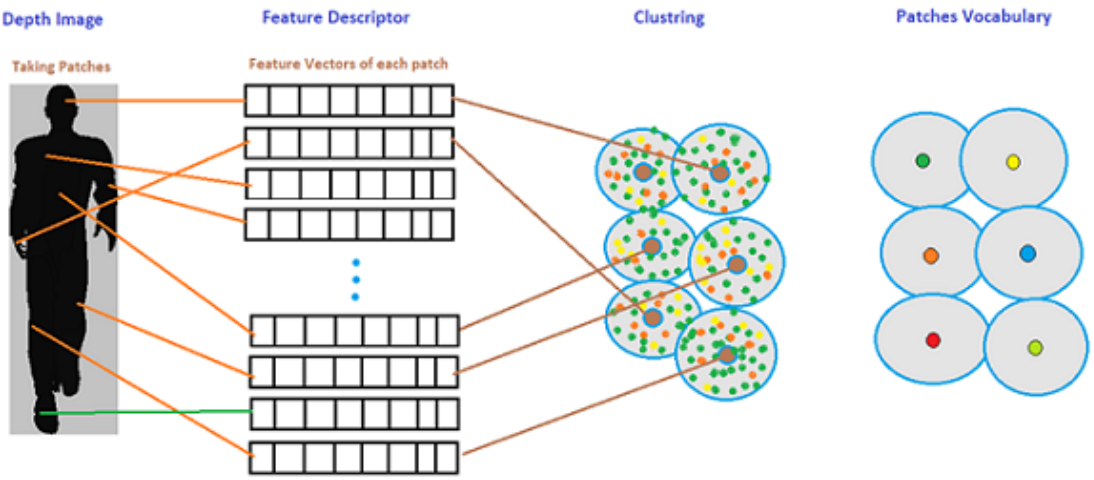


Figure 3.13: Overview of Bag of features technique

3.2.2 Local Features

Local features are used to find out the distinct pattern or structure in a image. This distinct pattern or structure can be recognized as image patch, edge and point. Usually local features are allied with image patch. The image patches differs each other from its adjacent patch by pixel intensities, pixel color and patch texture.

We used eigen values and eigen vector to extract local features form the patch. Figure 3.14 shows the graph that present the eigen values of the patch. In figure 3.14 the eigen values from 0 to 5 in x-axis are the most dominat eigen values and the rest of the eigen values are 0. To get rid of 0 eigen values we decided to take first 80% of eigen values as our local features for each patch.

We trained the random forest classifier using these local features. Given a test image we applied the same procedure as defined above and given to random forest classifier to predict the correct part of human body.

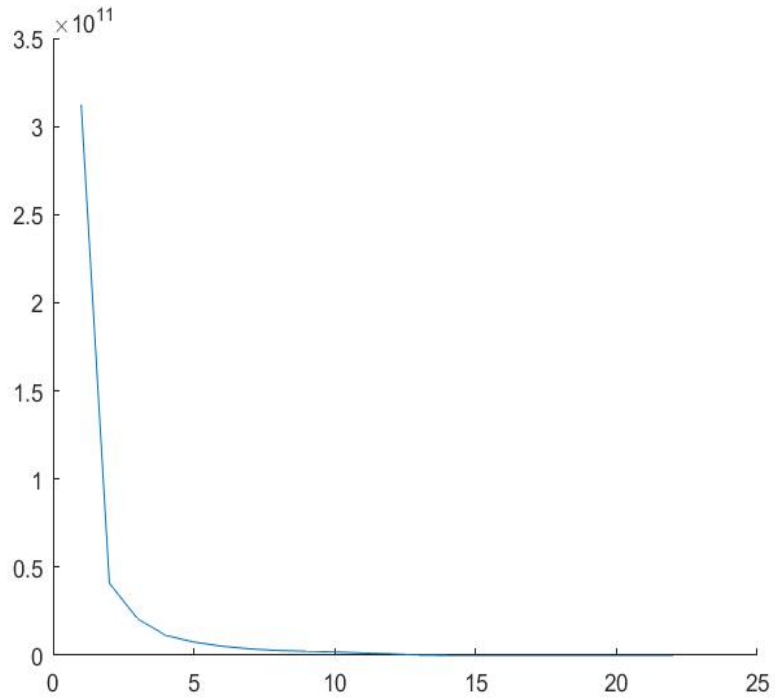


Figure 3.14: Patch based local features

3.3 Fusion of Features

Every feature we used in our thesis has different characteristics. Therefore we decided to fuse features together to check which feature is dominant. In pixel based method we fused depth and computer vision features together. We also fused the depth feature with SIFT and HOG as well to check dominance of depth feature with computer vision feature techniques or vice versa.

3.4 Feature Vector

As we are dealing with depth image in our thesis. In depth image there can be a possibility of two or more parts may have same depth values because of having

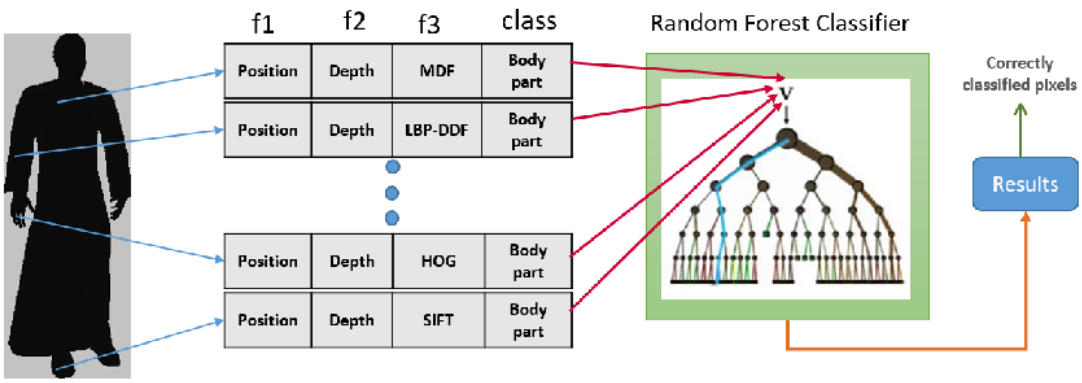


Figure 3.15: Feature vector for pixel based method

same distance from the depth sensor. However, If we apply feature extraction technique in one part we can have same feature for other part as well. The classifier may not perform well under these situation.

Therefore, we decided to use classified pixel position/location and its depth value as feature with other features. Using this we can easily differentiate the same depth values problem. Figure 3.15 shows the representation of feature vector in pixel based method.

3.5 Classifier

3.5.1 Decision Trees

Decision tree is a simple decision model that describe a hierarchical decisions and their significance. This model is widely used from last few decades [70]. it is very different from traditional tree but it produced much impressive results on previously unseen data. This model also known as generalization[71][72].

A simple tree is organized in a hierarchical structure with the collection

of nodes and edges. The nodes are further divided into child nodes and leaf node(terminal node) based on the certain split mechanism. A simple trees can be split into binary where each parent node have maximum two child nodes. It can also be split into multiple nodes where each parent node have more than two nodes. Figure 3.16a shows the example of binary trees where circles represents the nodes or internal nodes and square boxes shows the leaf nodes. The leaf nodes contains the scalar value and its normally consider a output values.

A decision tree is similar to binary trees in terms of hierarchical structure. However, it act as different than simple binary trees. It is used for decision making where each node is involved in decision based on some test function. The decision function is applied on every internal node until the it gets to a terminal node (predictor node) where the final result is stored. A decision tree is used for the classification and regression purposes where we need to predict the final output value of sample or instance class. Figure 3.16b shows the example of simple decision tree where it predicts whether the picture is captured in a outdoor scene or indoor scene. We can observe from figure 3.16b that every node is involved in decision function.

Classification in Decision Trees

classification of decision trees can be divided into training and testing phases. The first phase is training phase where the actual decision tree is formed. Given a training feature set a subset of training features are selected randomly at each node. Every node in decision tree is branch to left and right based on the decision

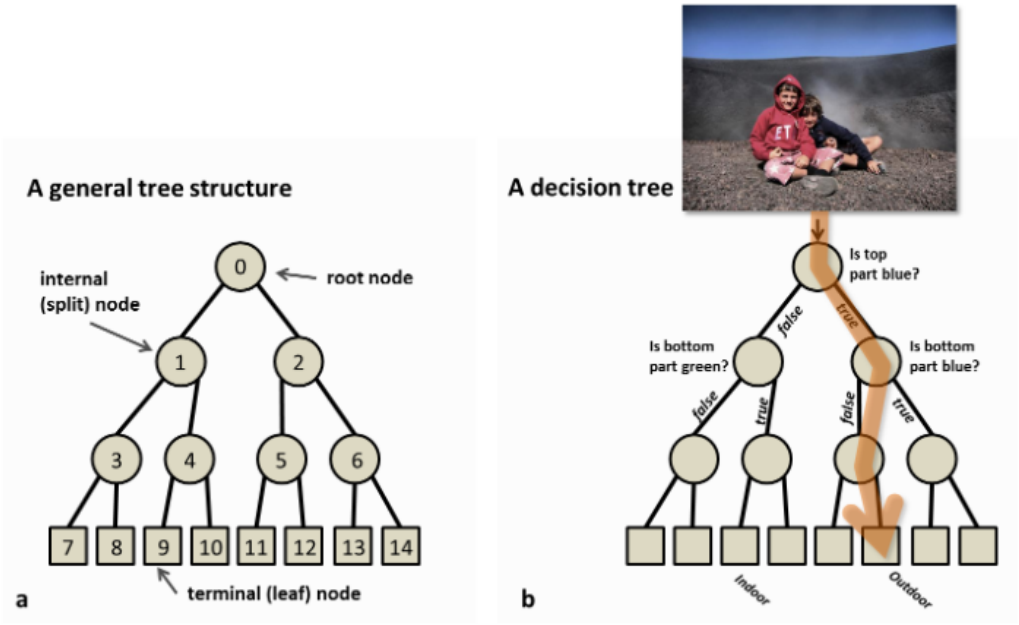


Figure 3.16: Simple and Decision trees[9]

at the node. Training a decision tree is a propagating process that every parent node passes the learned knowledge to the internal nodes. The nodes are in decision trees is represented a breadth-first order. Figure 3.17 shows the example of training tree. Figure 3.17a shows the training feature samples of different labeled data. Figure 3.17b shows the decision tree who's input is the subset of training features. Each node is branch to left and right until it meets some stopping criteria.

Once the decision tree is built from the training features or sample data. The testing features or sample data are given to the decision trees. The testing features are previously unseen data which is applies hierarchical in a decision tree. Start from root node each node applies its decision function to testing features. According to the results the test data is traverse down to left or right child until

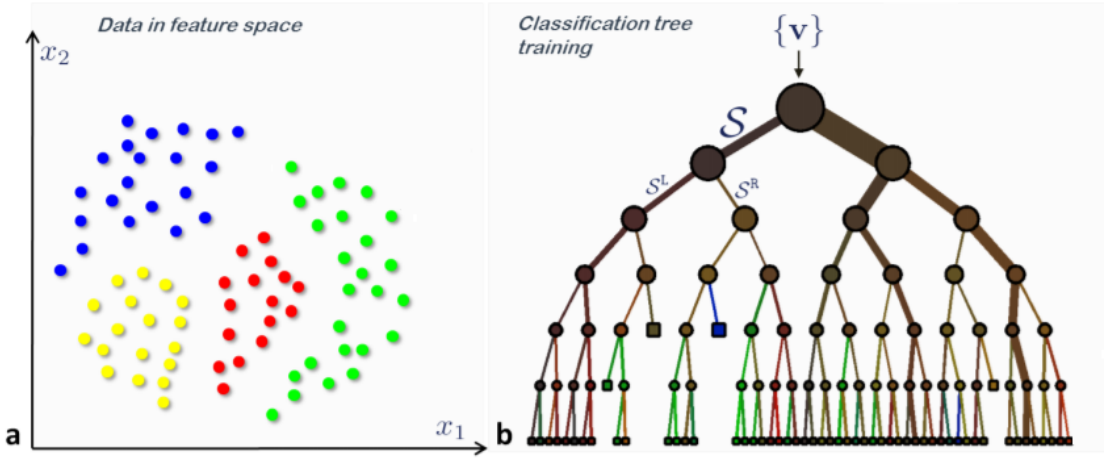


Figure 3.17: Training in Decision Tree[9]

it gets the final prediction which is held on the leaf node.

Impurity Measures

The nodes in the decision trees are split based on some splitting function or criteria. The best splitting criteria is one that gives the best informative split or improve the performance. There are many univariate splitting criterias but most common are Gini index and information gain [73] [74].

To best split the random features the information gain scoring function gives a quantitative measure of how much uncertainty is reduced by splitting a node according to particular attribute. The information gain is calculated by determining the difference of the parent Shannon entropy and the weighted sum of its children's entropy.

$$Gain(D, D') = H(D) - \sum_{s \in \{left, right\}} \frac{|Ds|}{|D|} H(D) \tag{3.5}$$

Where

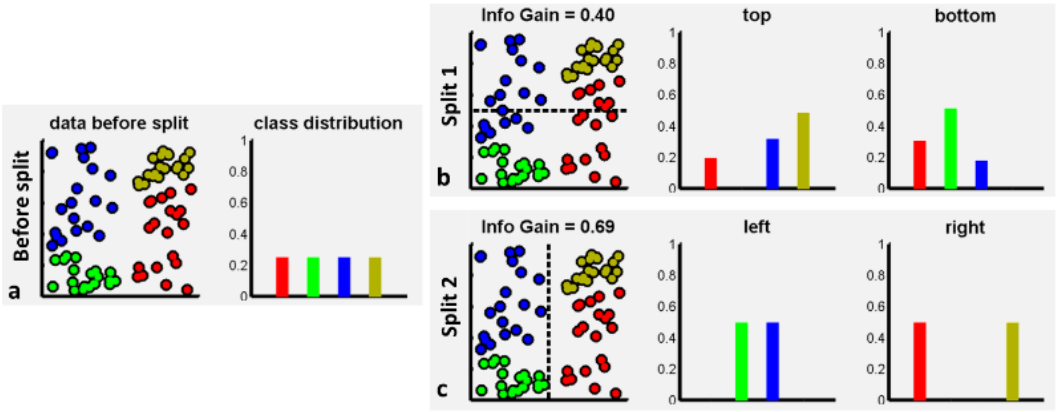


Figure 3.18: Information Gain before and after split[9]

$$H(D) = -\sum_{c \in C} \rho(c|D) \log_2(\rho(c|D)) \tag{3.6}$$

c is a target class and H(D) is the entropy in the class distribution of a certain node. Figure 3.18 shows the example of information on certain dataset points. Figure 3.18a shows the data points before the splits. The class distribution is uniform because all the class same number of data points. In figure 3.18b if we split the data points horizontally we got 0.40 information gain. However, if we split the same data points vertically in figure 3.18c. We got 0.69 information gain score. The max information gain score gives the best split. Therefore, we split the data points vertically in that case.

Stopping Criteria

The growing of the decision tree is recursively repeated until stopping condition is met. However, there are three types of stopping condition in which the decision tree is stop growing and stored resultant prediction at terminal or leaf node. The

Stopping condition are following.

- The maximum depth of decision tree is achieved.
- All instances in the training set belongs to a single value.
- The low information gain score which does not meet a certain threshold.

3.5.2 Random Forest

The Random forest [75] classifier has a fruitful history in computer vision applications. It is considered an important technique in machine learning. When it comes to performance, it outperforms most of the state-of-art machine learning classifiers especially when working with high dimensional data. It consists of multiple decision trees and has the ability to deal with multiple class problems.

Random forest is the collection of decision trees. According to [76] [77] [78] the testing accuracy increased if the number of decision trees increased monotonically. Over fitting is one of the major problem in classification model. However, in random forest model if we have enough number of decision trees the random forest classifier take care of it and won't over-fit the model [2]. The random forest classifier accumulates votes from other base classifier in order to improve the overall accuracy. The majority voting is used for the final classification.

Given a test instances to random forest model. The test instances traverses down from root node to leaf node through each decision tree in the model. The model produces a probabilistic prediction at the leaf node of each decision tree.

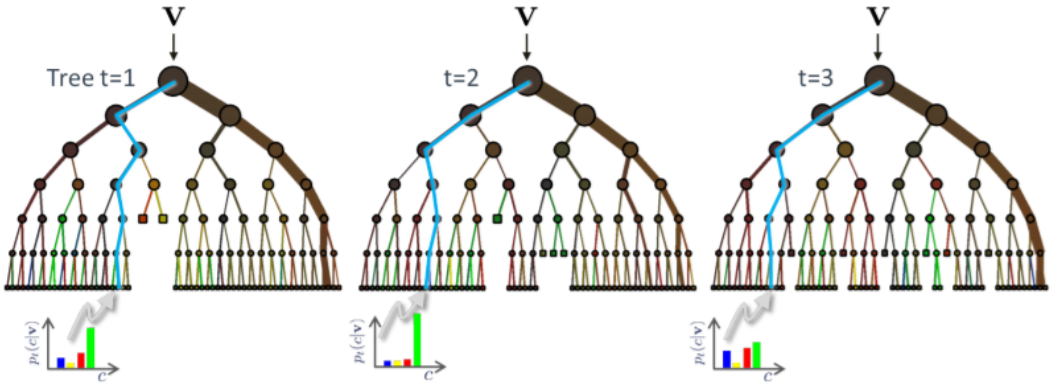


Figure 3.19: Random Forest testing[9]

Each leaf in a decision tree yields a posterior probability of class given test instances. The final output of predicted class is the average of all the decision trees posterior probability in the forest model. Figure 3.19 is an example of random forest testing. The model is consist of total 3 decision trees and V is the test instances that supplied to the trained decision trees. The final prediction of the class is calculated by:

$$\rho(c|v) = \frac{1}{T}\sum_{t}^{T} \Pr(c|v) \tag{3.7}$$

Where c is the predicted class, T is the total number of decision trees in the random forest model and v is the test instances.

Due to the randomization, fast training and testing. We used random forest as classifier in our thesis. In pixel based approach we have a huge data around 10 million pixels for training data and 60 million pixels for testing data. Due to this large training and testing data we restricted our random forest model with 3 decision trees. In order to increase the number of decision trees in the pixel based random forest model required high performance system.

However, in patch based approach where we used patches instead of pixels. The amount of training and testing data is comparatively less than pixel based approach. Our patch based random forest model consist upto 40 decision trees.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Datasets

There are no any datasets available online that deals with draped clothes like Arabic thobe, sub-continent dress etc. Therefore, in the thesis we used Ridwan [3] datasets for Arabic Thobe. Its a synthetic data and contains ground truth and depth image of same size and resolution.

In Thobe dataset we have two types of images. These two images consists of depth and ground-truth image. Both image have same size and resolution. The depth images contains the depth value which basically represents a distance from depth sensor. The range of depth value is between 0 to 4000mm. the background has constant value which is 50000. Where as the ground-truth image is consider as labeled RGB image which give us an information about pixel that from which body part the pixel is belong to. Table 4.1 shows the thobe dataset parameters and figure 4.1 shows the reprsentation of ground truth and depth of Thobe dataset.

Table 4.1: Dataset Parametersn

Thobe Dataset Parameters	
Training Data	16,000 Image
Testing Data	3,000 Images
No of Body Parts	26 Body Parts

Figure 4.1: Thobe depth and ground truth image dataset representation

4.2 Performance Metrics

Precision, Recall, and Accuracy

We used precision and recall metrics to evaluate the performance of every single body part in a human body. However, in some cases high precision is required whereas in some cases recall required to be high. In this thesis we used both and combined them to have a single score. We used F1-Score Metric which is the harmonic mean of both precision and recall. The harmonic mean in F1-Score gives appropriate score then arithmetic mean. These metrics are calculated as follows.

$$Precision(P) = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{4.2}$$

$$Accuracy(F1 - Score) = \frac{2PR}{P + R} \tag{4.3}$$

Where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

4.3 Reproduced Papers Results

4.3.1 Real-time human pose recognition in parts from single depth images

This paper is written by Shotton *et al.* [2]. The Shotton approached was rooted on object recognition strategies. They identify the human body parts and localize the 3D joint position. The machine learning classification techniques called Random Forest classifier was used for training and prediction. The Shotton approach was specifically designed and tested for western clothes like jeans, sports trousers, casual and dress shirts and it is invariant to the body, shape etc.

The authors does not public their dataset and other supporting material. Therefore, we reproduced their technique in our Thobe dataset. The nature of both database is same because both have depth image and ground truth images only difference is the dress. We wanted to check how Shotton *et al.* [2] simple depth feature works on thobe dataset. Table 4.2 shows the Shotton *et al.* [2] reproduced reults.

Table 4.2: Shotton *et al.* [2] Reproduced Results

Shotton Reproduced Results	
Total images	6000
Random no of features selected	2000
Depth	20
No of trees	3
Training images	80%
Testing images	20%
Accuracy:	40%

Table 4.3: Ridwan [3] Reproduced Results

Ridwan Reproduced Results	
Total images	20000
Random no of features selected	800
Depth	20
Region size	3
offset distance	300
No of trees	3
Training images	16000
Testing images	3000
Accuracy:	43%

4.3.2 Pose estimation of human wearing Thobe using depth images

Ridwan [3] done this work in his MS thesis.The approach he followed is same as Shotton *et al.* [2] approach with minor changes. Instead of taking two random offsets(u, v) points he takes two random regions and take the average of regions as offset(u, v) in a depth image. The other difference is that he only took 800 features whereas Shotton *et al.* [2] took 2000 features. Table 4.3 shows the Ridwan [3] reproduced results.

Table 4.4: Pixel Based Configurations

Configurations	
Total images	19000
Random no of features selected	800, 1000, 1200, 1400, 2500
Depth	20
No of trees	3
Training images	16000
Testing images	3000

4.4 Pixel-Based Approach Results

In our framework we applied different feature extraction technique for recognizing human body parts but we end up a technique which gives best results among all other techniques. Table 4.4 shows the training and testing parameters.

4.4.1 Median Depth Features Results

We implemented this feature using three different ways.In first two cases we fixed the windows size to 3x3, 5x5 and 7x7. However, the background pixels can be important factor for effecting the accuracies up and down. Therefore, in first case we included background pixels only if our fixed sized windows contains background pixel. Whereas, in second case we avoided the background pixel by replacing the background value with 0. In third case we used randomization technique. Instead of fixing windows size we generated window height and width randomly. The maximum size of window is 3x3. Table 4.5 shows the results of median depth features.

From table 4.5 we can observed that random window size produced better results than other fixed window size not only on 800 pixels but it is also leading

Table 4.5: Median Depth Features Results

	Background Included			Background Not Included			Mixed
Pixels	**3x3**	**5x5**	**7x7**	**3x3**	**5x5**	**7x7**	**Random**
800	42.82%	42.82%	42.81%	42.74%	42.58%	42.57%	**43.08%**
1000	42.98%	43.00%	42.99%	42.91%	42.92%	42.92%	**43.31%**
1200	43.12%	43.03%	43.20%	43.14%	43.15%	43.14%	**43.37%**
1400	43.17%	43.33%	43.33%	43.31%	43.29%	43.29%	**43.40%**
2500	43.78%	43.78%	**43.79%**	43.75%	43.74%	43.74%	43.60%

up to 1400 pixels. However, window with size 7x7 including background pixels performed better on 2500 pixels. Inclusion of background pixels in a window have a better accuracy rate as compared to the results of non background pixels. Therefore, we can say that the background pixels have a little positive influence on the pixel accuracies. It is also observed from table 4.5 that if we increase the number of pixel the acuracies will also increase. We increased almost 3 times of the pixels and we improved total 1% additional accuracy in all three cases.

Results in the table 4.5 were executed only once. However, by observing the accuracies in table 4.5. We noticed that for 2500 pixels the results are better as compared to other chunks of pixels. The difference of accuracies among the window sizes for 2500 pixels are also very minimal. Therefore, we decided to execute every single window for 2500 pixels up to 30 times and used the average as an accuracy. The purpose of this is to calculate the margin of error using confidence interval. The confidence interval gives us confidence that our results lies within the lower and upper interval.

Figure 4.2 shows the average results of 30 runs for different window sizes for 2500 pixels. All the techniques (background, without background and mixed)

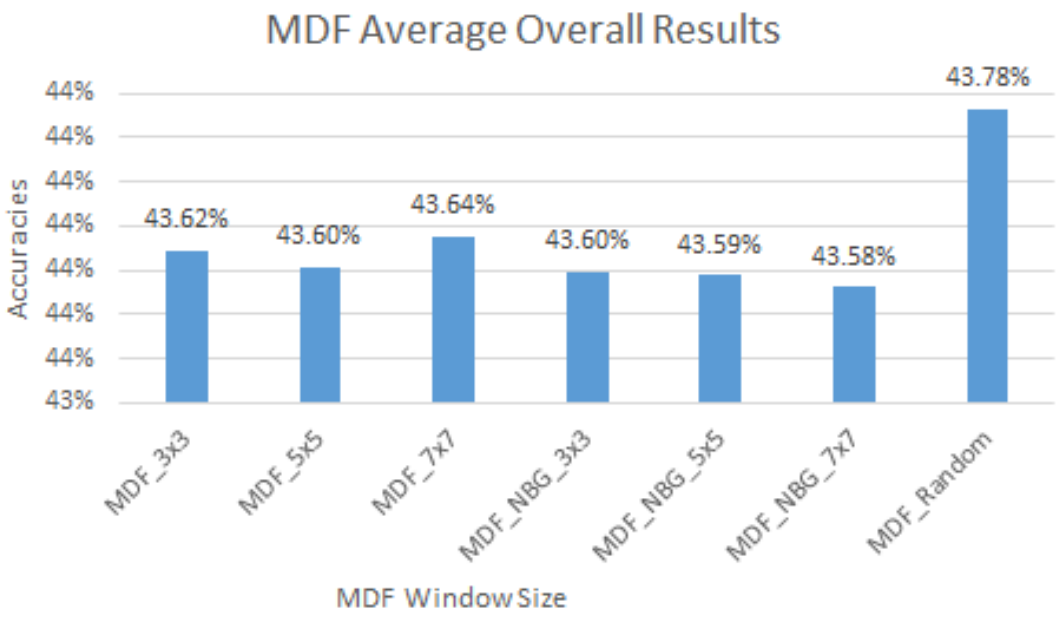


Figure 4.2: MDF Average Results for 2500 pixels

produced almost same results. But random window in mixed technique leading with 43.78% accuracy. However, if we compared these average results with single run in table 4.5. We can observed that the accuracies of fixed windows declined in average results.

Similarly, figure 4.3 showing the 95% confidence interval. We observed from figure 4.3 the margin of error is very low. However, the confidence interval of fixed windows in both techniques (background included and background not included) is overlapping to each other. Therefore we can say that the accuracies of fixed windows for both techniques are significantly same. Random windows in mixed technique is the only one that significantly different from fixed window sizes. In conclusion to median depth feature we would say that the random window size feature is better than others fixed window size.

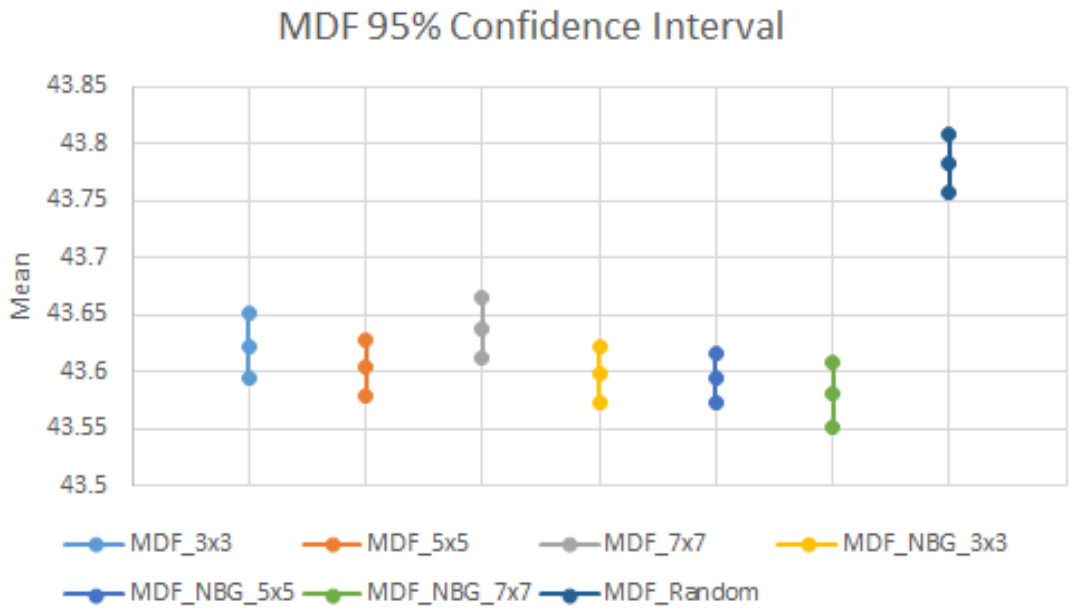


Figure 4.3: MDF 95% Confidence Interval Results

Individual Body Parts Results

The accuracies of individual body parts in pixel based approach helps to figure out that how many pixels are correctly classified individually. We ran experiments up to 30 times and table 4.6 indicates the average accuracies of each body part for 2500 pixels. In table 4.6 the first three fixed windows (3x3, 5x5 and 7x7) included the background pixel. However, the other three fixed windows the background pixels not included, The "R" in table 4.6 represents the random window size.

Random window in table 4.6 overall produced better accuracies in majority of the body parts when compared to fixed window sizes. Out of total 27 body parts random window produced good results in 11 body parts. Similarly, the fixed window with background pixels included also produced better accuracies in 11 body parts. However, when we compared the fixed windows together while having an

involvement of background pixel. We came up that the fixed window with background pixel included producing slightly better score then the one doesn't have the background pixel. There are only 4 body parts where the accuracies are better than others in the absence of background pixels in fixed windows. Therefore, we can sum up that the background pixels have impact on the accuracies of recognizing human body part. The inclusion of background pixels in the window helps to improve the accuracy of a certain body part.

The highest accuracy we got in table 4.6 is 54.47% which is "right_head". The lowest is 7.34% which is "right_arm3". The body parts "right_arm3", "left_arm3" and "left_ankle" is rarely visible in the image. Most of the time these body parts occluded by a human pose. Sometimes the random pixel hits is too small to hit these body parts spots specifically. So that's why the accuracies of these body parts are too low. However, in the patch based results section we also compared the accuracies of these body parts with patch based approach to differentiate which approach is better to deal with these kinds of body parts accuracies. Figure 4.4 shows the graphical representation of each body part accuracy in median depth technique for 2500 pixels.

Upper and Lower Body Parts Results

Figure 4.5 shows the median depth feature upper and lower body parts score for 2500 pixels. These average classes accuracies are calculated from confusion matrix using equation 4.3. Figure 4.5a shows the upper body parts accuracy. We observed that all the techniques whether fixed window or random window have almost same

Table 4.6: MDF Individual Body Parts Results for 2500 Pixels

	right_leg	**right_chest**	**thobe_part**	**right_arm1**	**right_arm2**	**right_abs**	**right_elbow**	**right_head**	**right_face**
3x3	42.82%	50.19%	48.76%	25.39%	29.81%	52.02%	37.02%	54.43%	52.22%
5x5	**42.86%**	50.29%	48.76%	25.36%	29.87%	51.98%	37.02%	54.33%	52.04%
7x7	42.84%	50.24%	48.74%	**25.58%**	29.83%	52.02%	37.01%	**54.47%**	52.19%
3x3	42.75%	50.26%	48.73%	25.52%	**29.91%**	51.93%	36.93%	54.16%	52.00%
5x5	42.79%	50.22%	48.74%	25.46%	29.81%	51.93%	**36.97%**	54.26%	52.10%
7x7	42.80%	50.26%	48.72%	25.31%	29.71%	51.95%	36.92%	54.30%	52.07%
R	42.55%	**50.44%**	**50.01%**	25.50%	29.25%	**52.86%**	36.70%	54.41%	**52.33%**
	right_wrist	**left_arm2**	**right_hand**	**neck**	**right_foot**	**right_arm3**	**left_foot**	**left_elbow**	**left_chest**
3x3	27.98%	**35.50%**	29.81%	**38.45%**	44.67%	7.16%	47.63%	36.16%	40.94%
5x5	27.93%	35.45%	29.78%	38.31%	44.64%	7.03%	**47.68%**	**36.17%**	40.70%
7x7	27.99%	35.48%	**29.88%**	38.34%	44.68%	7.11%	47.64%	36.14%	41.01%
3x3	27.89%	35.37%	29.72%	38.30%	44.56%	**7.34%**	47.65%	35.94%	40.97%
5x5	27.91%	35.36%	29.64%	38.36%	44.59%	7.30%	47.62%	36.00%	40.92%
7x7	27.88%	35.31%	29.65%	38.33%	44.68%	7.33%	47.65%	35.97%	40.80%
R	**28.42%**	35.13%	29.58%	36.50%	**44.89%**	6.76%	47.29%	35.94%	**41.24%**
	left_arm1	**left_arm3**	**left_abs**	**left_leg**	**left_face**	**left_head**	**left_hand**	**left_wrist**	**left_ankle**
3x3	29.95%	17.17%	41.76%	42.93%	48.92%	53.46%	29.74%	25.37%	7.49%
5x5	29.96%	17.11%	41.79%	42.84%	49.12%	**53.54%**	29.77%	25.38%	7.59%
7x7	**30.15%**	**17.17%**	41.77%	42.92%	49.01%	53.45%	29.79%	**25.45%**	7.54%
3x3	30.02%	17.08%	41.82%	42.96%	**49.21%**	53.54%	29.67%	25.19%	7.54%
5x5	30.08%	17.15%	41.84%	42.92%	49.15%	53.46%	29.55%	25.20%	7.73%
7x7	29.92%	17.13%	41.72%	42.94%	49.00%	53.43%	29.63%	25.28%	7.66%
R	29.04%	16.41%	**42.12%**	**43.09%**	47.34%	51.14%	**30.74%**	25.21%	**8.59%**

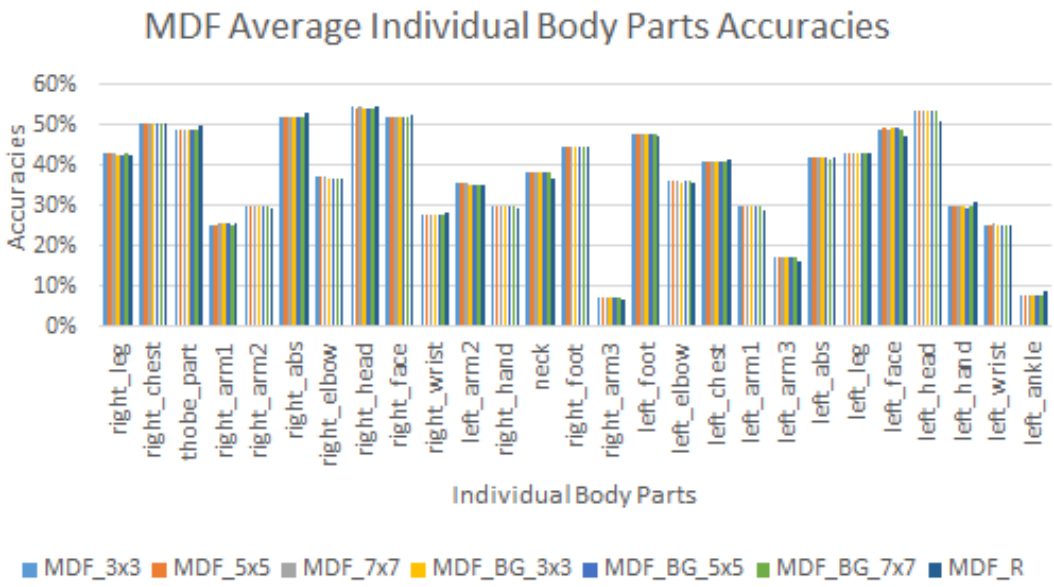
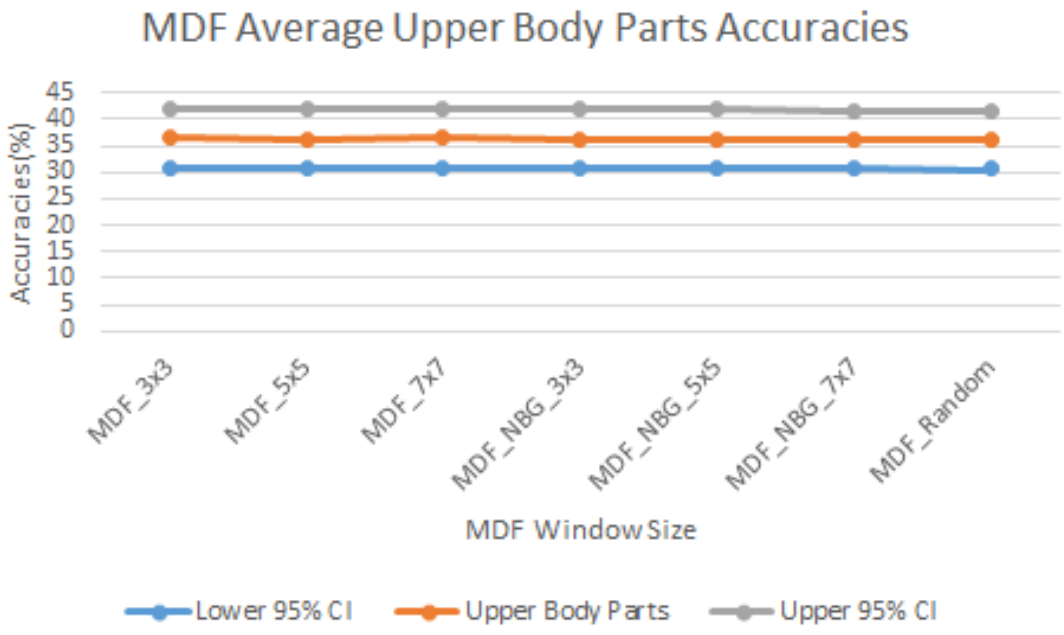


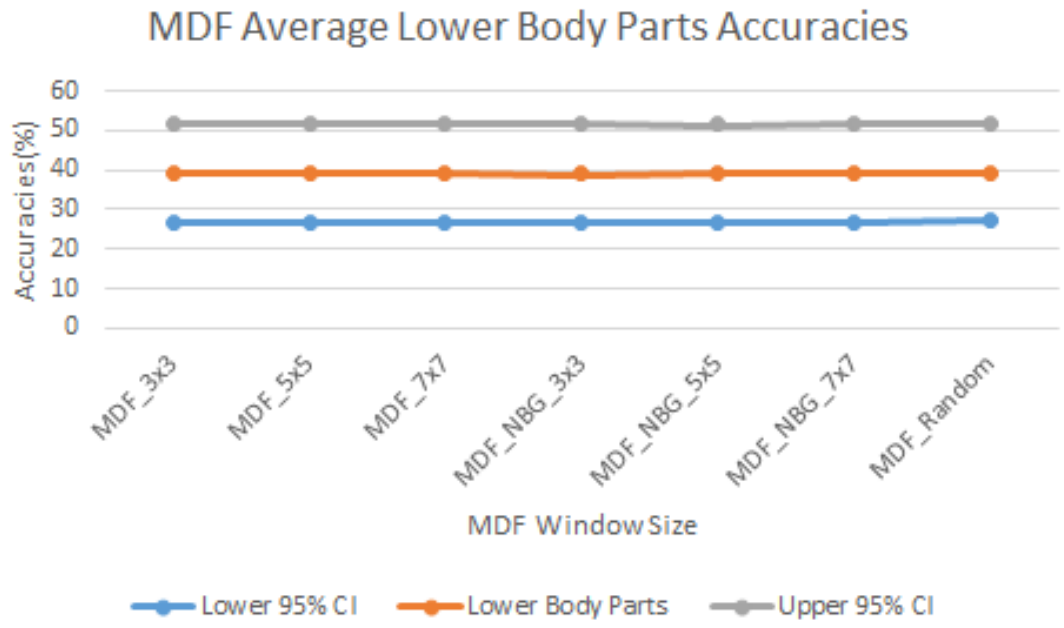
Figure 4.4: MDF Average Individual Body Parts Accuracies for 2500 Pixels

upper body accuracy. The difference of accuracies among all techniques is very minimal so it is difficult to say that one technique is taking dominant lead over others.The maximum accuracy we got in upper body is 36.38% and the lowest is 36.05%.

We have noticed the similar case in lower body parts accuracies as well. Figure 4.5b shows that all the techniques scored almost same with very minor difference in accuracies. There is no clear winner here as well. The random window have the maximum accuracy with 39.40% and fixed window 3x3 without background included have the lower accuracy rate with 39.04%. All the lower body parts scored over 43% except the "left_ankle"body part. The body part "left_ankle" has lowest accuracy among all other body parts. This body part is most of time is absent in the image. Out of every 100 to 200 images approximately this body part appeared only once. Sometimes it never gets maximum number of random



(a) MDF Average Upper Body Parts Accuracy for 2500 Pixels



(b) MDF Average Lower Body Parts Accuracy for 2500 Pixels

Figure 4.5: MDF Upper and Lower Body Parts Results

spots that's why this body part has lower accuracy as compared other body part.

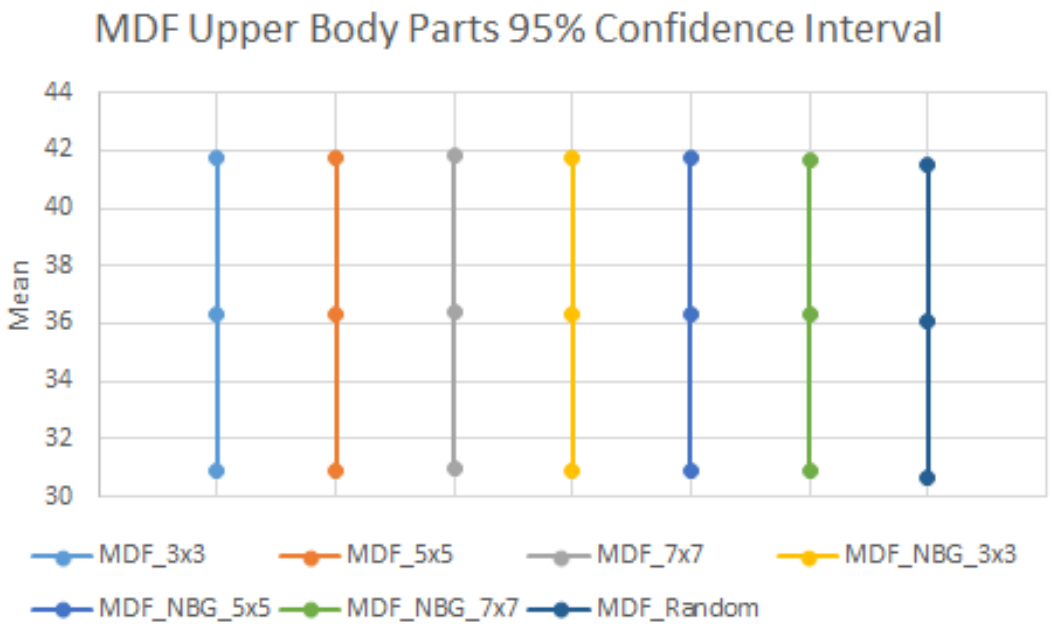
However, to make sure that all window based techniques in median depth feature have same upper and lower body parts score. We calculated the confidence

interval of upper and lower body parts. Figure 4.6 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.6a and figure 4.6b indicates that the mean of all the techniques are overlapping to each other in both upper and lower body parts. Therefore, we can conclude that all techniques in median depth features are significantly same for 2500 pixels.

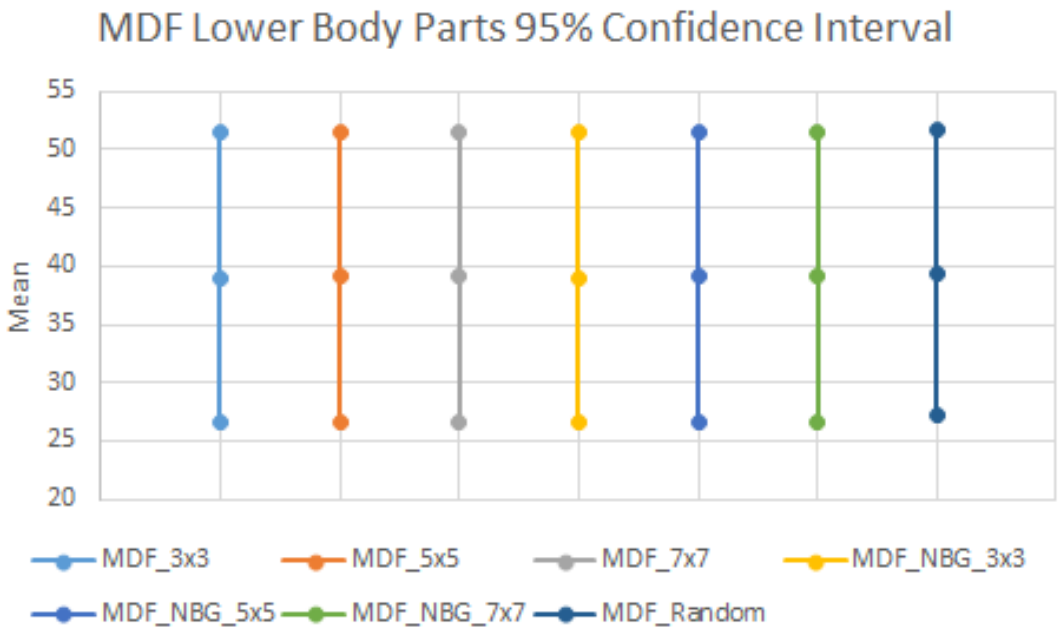
4.4.2 LBP-Decimal Depth Features Results

In decimal depth feature we fixed the window size to 3x3. This 3x3 window gives us a 8-bit decimal number and this is our feature for the classified pixel x. We executed the experiments up to 30 times on different chunks of random pixels as described in table 4.4 and calculated the average. Figure 4.9 shows the average of LBP-DDF accuracies from random pixel 800 to 2500. We observed that random pixels have a positive relation with accuracy. For example, when we increased the number of random pixels we noticed some slight improvement in the accuracy. We almost got 1% increase in the accuracy from 800 random pixels to 2500. However, we improved almost 4 % accuracy as compared to median depth technique. The highest accuracy we got in decimal depth technique is approximately 48% for 2500 chunk of random pixels.

Figure 4.8 shows the 95% confidence interval results. We noticed that all the random chunks of pixels have very low variance. It is because of the higher population size. In our case the population size we used is 30. We can say that we are 95% confident that the accuracy of each chunk of random pixel lies between



(a) MDF Upper Body Parts 95% Confidence Interval for 2500 Pixels



(b) MDF Lower Body Parts 95% Confidence Interval for 2500 Pixels

Figure 4.6: MDF Upper and Lower Body Parts Confidence Interval Results

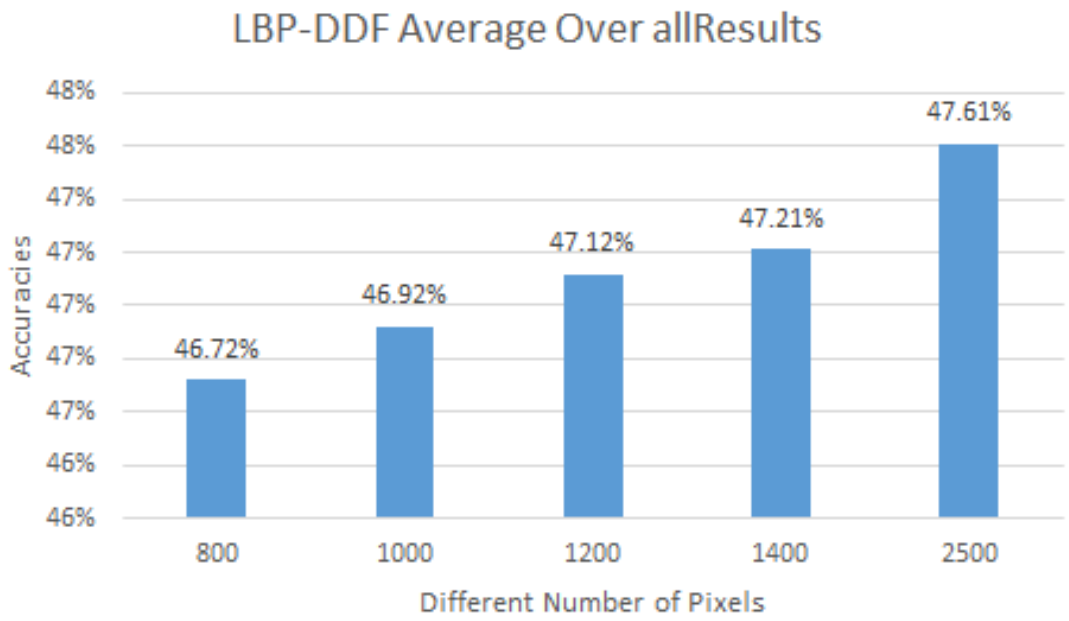


Figure 4.7: LBP-Decimal Depth Features Overall Results

the lower and upper confidence limit as mentioned in a figure 4.8.

However, if you look at the overlapping of ranges of different chunks of random pixels in figure 4.8. We figure it out that none of any random chunk of pixels intervals are overlapping. The difference of upper and lower interval limit between 1200 and 1400 chunk of random pixels is very small but they are not overlapping. Therefore, we can say that the accuracies of all the random chunks pixels are statistically not significant. In other words every time we run the experiments all the chunks of random pixels will get the different accuracies.

Individual Body Parts Results

Table 4.7 represents the accuracy of 27 body parts. The results are not very impressive for few upper body parts. The majority of body parts scored under 50%. The 800 random pixels performed worst as compared to 2500 random pixels

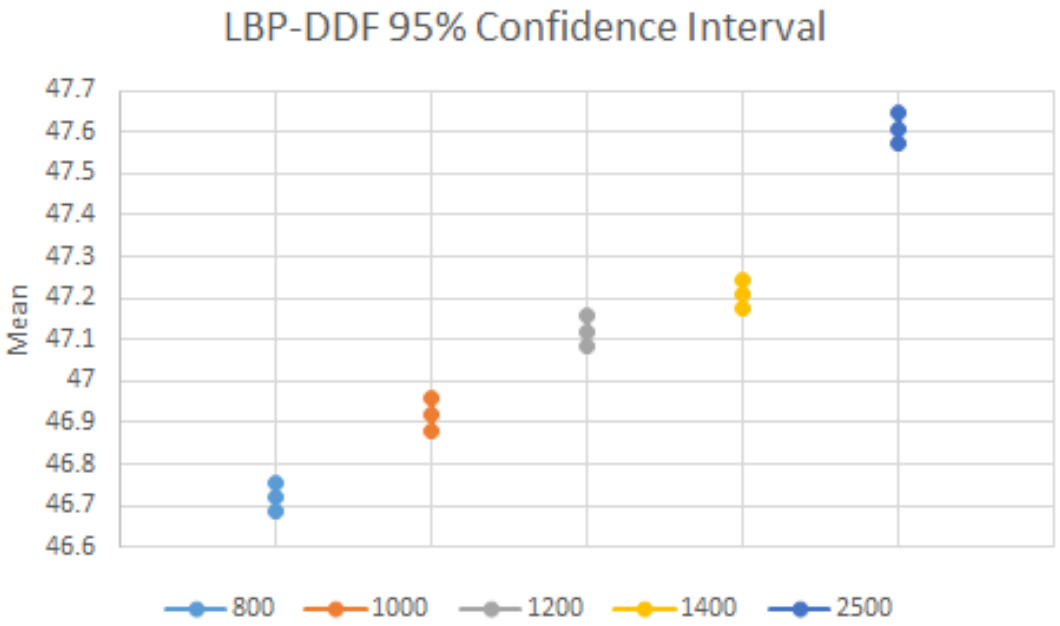


Figure 4.8: LBP-Decimal Depth Features 95% Confidence Interval Results

no single body part for 800 random pixels performed better as compared to others.

However, when we increased the random number of pixels upto 2500 we noticed that majority of the body parts accuracy improved. We also observed the body part which are bigger in size have better accuracy than the smaller ones. The "right_face" body part having a highest accuracy in decimal depth technique followed by "left_head" body part with 58.36% and 57.69% respectively. The lowest accuracy we got in decimal depth technique is the "right_arm3" and "left_ankle" with 9.66% and 12.26% respectively. The reason of having low accuracy in these both parts because of smaller in size and have less number of random pixels hits as compared any other part in the body. Almost all the lower body parts scored over 45% and 3 of them scored over 50%. Figure 4.9 shows the graphical representation of each body part accuracy in decimal depth technique.

When compared to median depth feature technique we improved the acuuracies

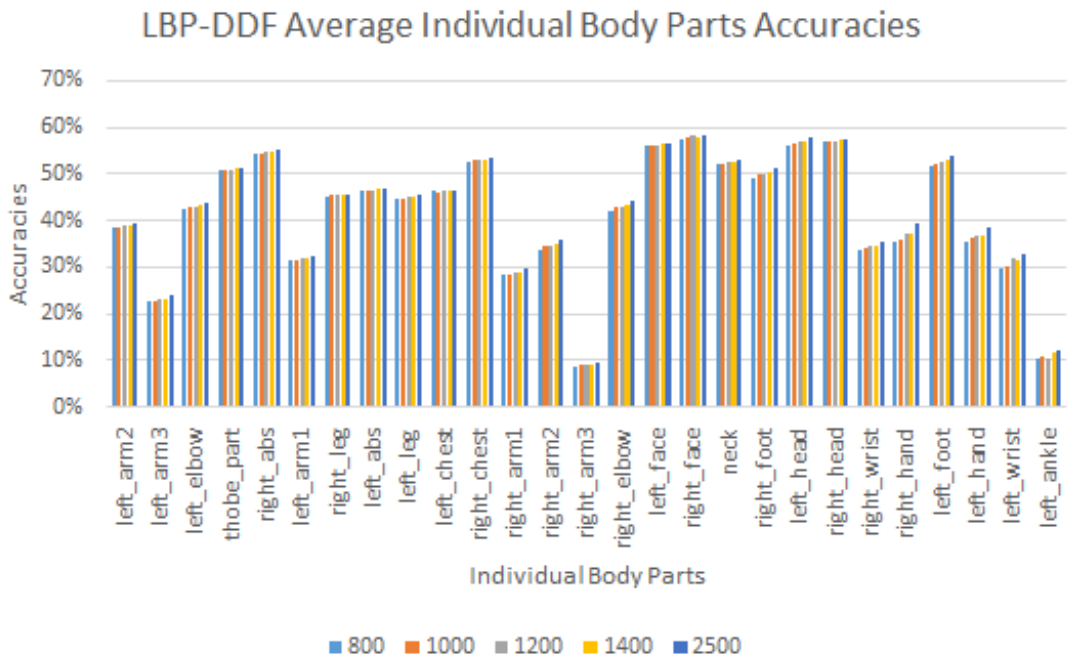


Figure 4.9: LBP-Decimal Depth Individual Body Parts Results

of almost all the body parts. In median depth feature technique only 6 body parts were able to score above 50%. Whereas in LBP-DDF technique we improved further 4 body. So, total 10 body parts scored over 50% in LBP-DDF technique. In both techniques none of the body parts scored over 60%.

Upper and Lower Body Parts Results

Figure 4.10 shows the upper and lower body parts scores for different random number of pixels. These average classes accuracies are calculated from confusion matrix using equation 4.3. Both upper and lower body parts scored below 50%. We improved only 2% accuracy for upper body parts from 800 random pixels to 2500 random pixels in figure 4.10a. The reason of having below 50% score in upper body parts is because of smaller body part e.g "left_arm1","right_arm1",

Table 4.7: LBP-DDF Individual Body Parts Results

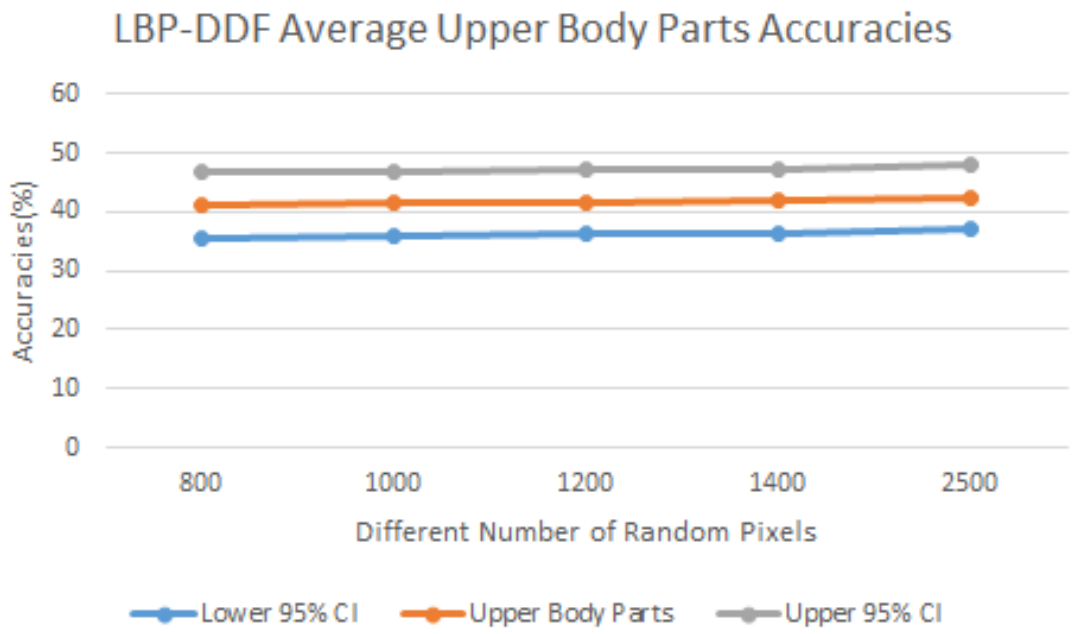
	left_arm2	**left_arm3**	**left_elbow**	**thobe_part**	**right_abs**	**left_arm1**	**right_leg**	**left_abs**	**left_leg**
800	38.47%	22.66%	42.32%	50.78%	54.41%	31.30%	45.17%	46.48%	44.66%
1000	38.74%	22.87%	42.79%	50.75%	54.57%	31.59%	45.44%	46.59%	44.72%
1200	38.90%	23.14%	42.94%	50.97%	54.83%	31.86%	45.37%	46.48%	45.08%
1400	38.99%	23.23%	43.26%	51.09%	54.76%	31.93%	45.52%	46.90%	45.07%
2500	**39.44%**	**23.84%**	**43.97%**	**51.38%**	**55.10%**	**32.31%**	**45.77%**	**46.95%**	**45.42%**
	left_chest	**right_chest**	**right_arm1**	**right_arm2**	**right_arm3**	**right_elbow**	**left_face**	**right_face**	**neck**
800	46.24%	52.72%	28.43%	33.91%	8.67%	42.07%	55.93%	57.61%	52.15%
1000	46.07%	53.10%	28.47%	34.44%	9.03%	42.71%	56.09%	57.87%	52.38%
1200	**46.64%**	52.98%	28.91%	34.61%	9.05%	43.02%	56.20%	58.13%	52.73%
1400	46.43%	53.01%	28.84%	34.85%	9.13%	43.35%	56.52%	57.79%	52.74%
2500	46.44%	**53.31%**	**29.54%**	**35.74%**	**9.66%**	**44.06%**	**56.62%**	**58.36%**	**53.20%**
	right_foot	**left_head**	**right_head**	**right_wrist**	**right_hand**	**left_foot**	**left_hand**	**left_wrist**	**left_ankle**
800	49.18%	56.06%	57.12%	33.74%	35.38%	51.64%	35.58%	29.63%	10.19%
1000	50.01%	56.61%	57.13%	34.28%	35.99%	52.19%	36.46%	30.38%	10.73%
1200	50.12%	57.00%	57.08%	34.37%	37.14%	52.65%	36.57%	31.85%	10.58%
1400	50.39%	57.05%	57.26%	34.74%	37.24%	52.85%	36.79%	31.61%	11.50%
2500	**51.42%**	**57.69%**	**57.63%**	**35.33%**	**39.40%**	**53.95%**	**38.60%**	**32.96%**	**12.26%**

"left_arm2","right_arm2", "left_arm3" and "right_arm3". These body parts got lower number of random hits as mentioned in figure 4.46 and that's why all these parts scored below 40%. It will be interesting to see the accuracies of these body parts in patch based approach.

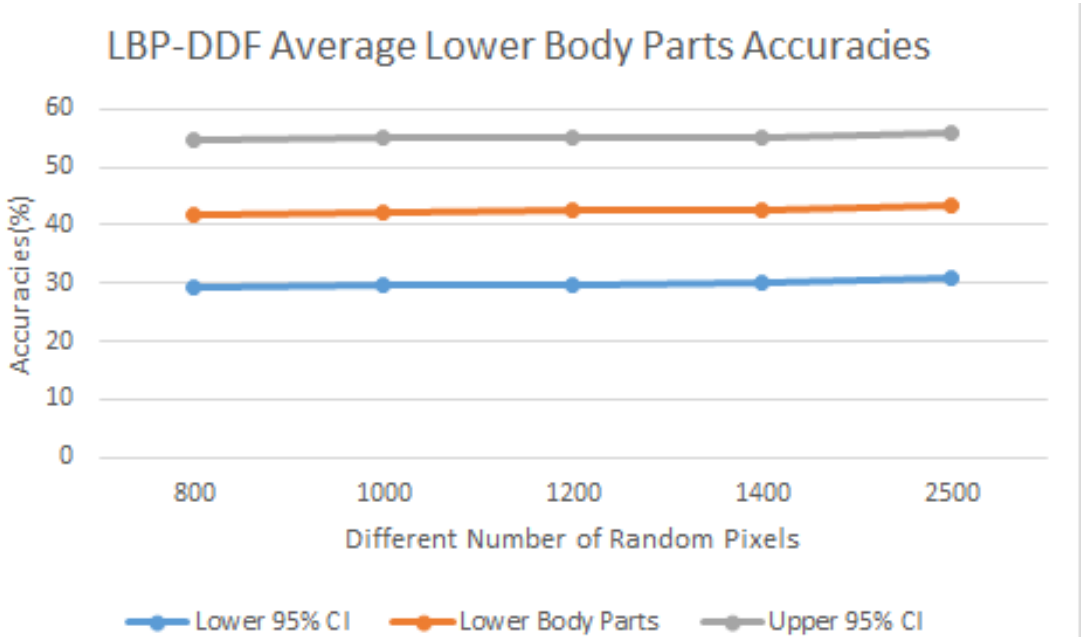
Similarly, The lower body parts also scored below 50%. Figure 4.10b shows that for every chunks of random pixels the average lower body parts improved only 1% than average upper body parts score. However, the average lower body parts improved approximately 2% scored when we increased the number of random pixels from 800 to 2500. The highest accuracy we got is 43.37% at 2500 random pixels and lowest is 43.94% at 800 random pixels. All the lower body parts scored over 45% except the "left_ankle" body part which scored only 12.26%. This body part is most of time is absent in the image. Out of every 100 to 200 image approximately this body part appeared only once. Sometimes it never gets maximum number of random spots that's why this body part has lower accuracy as compared other body part.

We calculated the confidence interval of upper and lower body parts. Figure 4.11 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.11a and figure 4.11b indicates that the mean of all the techniques are overlapping to each other in both upper and lower body parts. Therefore, we can conclude that we are 95% confident that the lower and upper body parts from 800 to 2500 random pixels in LBP-DDF technique are statistically significantly.

In comparison with median depth feature technique. In LBP-DDF we im-



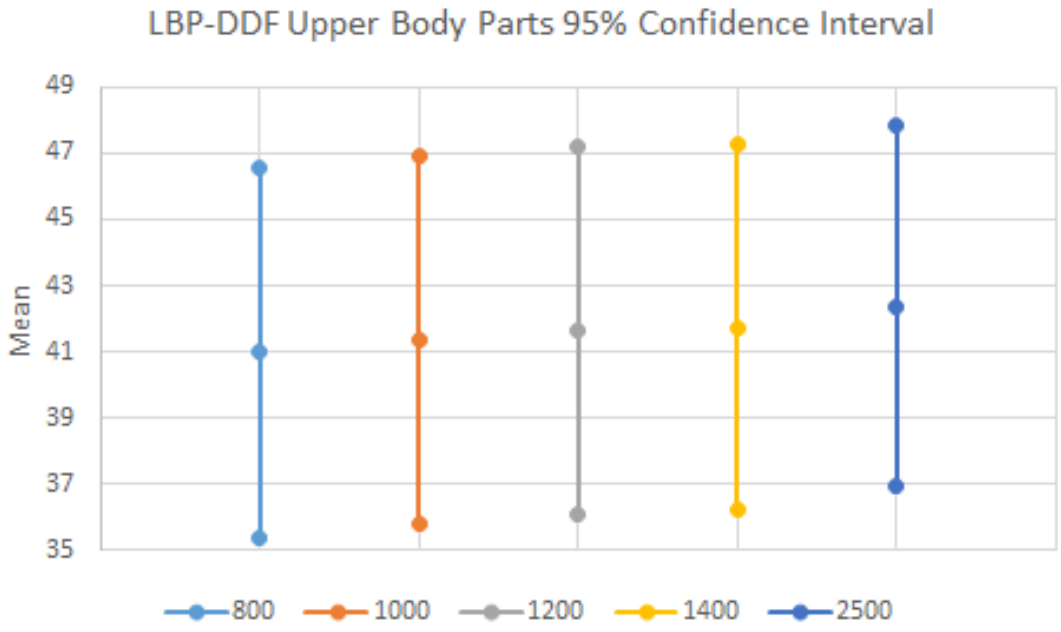
(a) LBP-DDF Average Upper Body Parts Accuracies



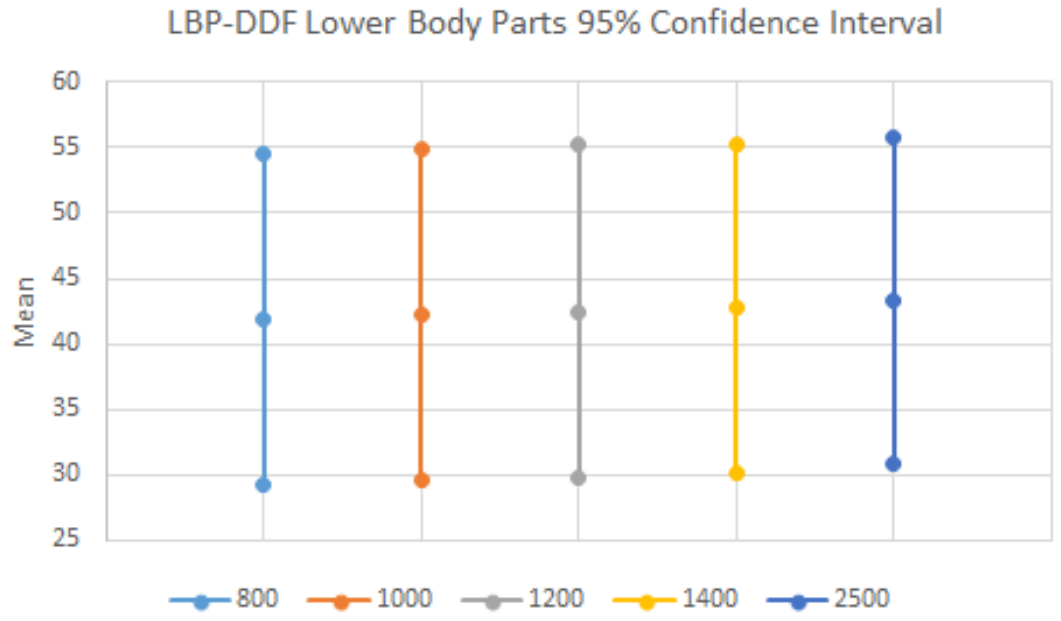
(b) LBP-DDF Average Lower Body Parts Accuracies

Figure 4.10: LBP-DDF Upper and Lower Body Parts Results

proved both upper and lower body parts scores. We improved 6% in upper body score and almost 4% in lower body score. Table 4.7 also indicated that we improved every single body part score from median depth features.



(a) LBP-DDF Upper Body Parts 95% Confidence Interval



(b) LBP-DDF Lower Body Parts 95% Confidence Interval

Figure 4.11: LBP-DDF Upper and Lower Body Parts Confidence Interval Results

4.4.3 HOG Results

After evaluating depth features we used computer vision feature extraction techniques to improve the results. The computer vision techniques are robust and highly capable of capturing a minor details in the region/area of interest. We used HOG in which the every random pixel is used to extract the features for a particular part. We used 2x2 block and the cell size is 8x8. According Dalal *et al.* [65] this is the best configurations for human detection. When a HOG window is out of bound of the image it produced null vectors. We removed all null vectors from the feature vector. Figure 4.12 shows the results of HOG descriptor in a depth image.

Likewise, previous techniques this one is also executed 30 times for each chunk of random pixels.HOG descriptor performed slightly better than depth based techniques. We improved 2% accuracy from LBP-DDF technique and almost 6% from MDF technique. However, similar to other depth techniques the accuracy is also increased when we increased the number of pixels from 800 to 2500 using HOG technique. At 800 random Pixels we got almost 48% accuracy compared to other previous techniques this best we got at 800 pixels. The HOG is gradient based descriptor so it means by applying the gradients on depth based dataset giving us a little edge over simple depth features for improving an accuracy. We improved almost 2% accuracy from 800 to 2500 pixels. The highest accuracy we got in HOG is 49.52% which approximately 50%.

Figure 4.13 shows the 95% confidence interval results. The population size

we used is 30. We observed from figure 4.13 that the confidence interval of 100, 1200 and 1400 random pixels are overlapping to each other. Their mean values are in the range of lower and upper limit of confidence interval. Therefore, we can say that we are 95% confident that the random pixels 100, 1200, and 1400 are statistically significant. This intuition make sense because the difference among the accuracies of these 3 random chunks of pixels are very close to each other. That's why we are 95% confident that these 3 random chunks of pixels produce same accuracy. We also noticed the confidence interval of 800 and 1000 random pixels. Their mean values are not overlapping in the range but their intervals are overlapping to each other. The difference is exactly 0.2% which is very minor so therefore these 2 random chunks of pixel considered statistically significant. The 2500 random chunk of pixels is different from other random chunks of pixels. The confidence interval of 2500 pixels is not overlapping any other random chunks of pixels. Therefore, we are 95% confident that 2500 random chunk of pixels is statistically not significant as compared to other random chunks of pixels.

Individual Body Parts

Table 4.8 represents the accuracy of 27 body parts using HOG technique. The results are not very impressive for few upper body parts. There are 14 body parts in this technique scored under 50% which is better than MDF and LBP-DDF technique where we recorded 20 body parts scored under 50%. The 800 random pixels performed worst as compared to 2500 random pixels no single body part for 800 random pixels performed better as compared to others.

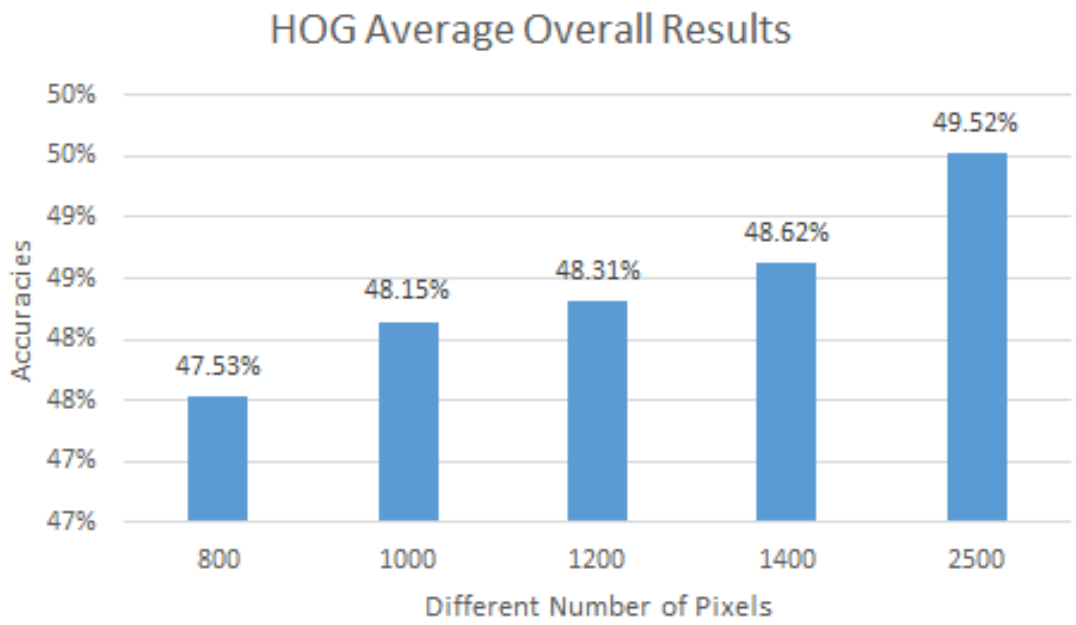


Figure 4.12: HOG Overall Results

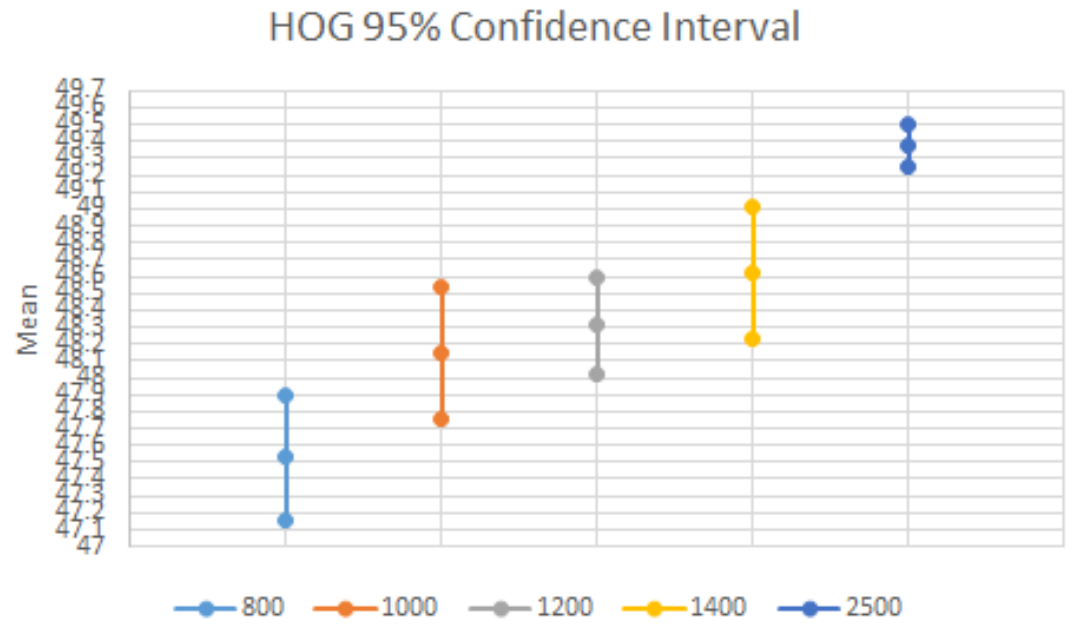


Figure 4.13: HOG 95% Confidence Interval

In HOG individual parts accuracy we have seen much improvements in almost every body part as compared to depth techniques in table 4.8. However, the LBP-DDF techniques have slightly better accuracy rate for "left_arm3", "right_abs", "right_leg", "right_arm1", "neck", "left_ankle" and "right_arm3" body parts than HOG. When we increased the number of random pixels the body parts accuracy also increased. We observed one thing common in depth and HOG techniques which is the "left_arm3", "left_ankle" and "right_arm3" body parts performed worst. The reason of having low accuracy in these both part because of smaller in size and have less number of random pixels hits as compared any other part in the body. The accuracy of these both parts is under 35%. The maximum accuracy we got is 66.21% for the "left_face" body part. This is a significant improvement as compared to depth techniques which was recorded 56.62% and 47.34% respectively. Almost all the lower body parts scored over 45% and 4 of them scored over 50% except "left_ankle". The "left_ankle" body part performing worst in depth based features and here in this technique as well. Figure 4.14 shows the graphical representation of each body part accuracy using HOG technique.

When compared to MDF and LBP-DDF technique we improved the acuuracies of 20 body parts. In MDF technique only 6 body parts were able to score above 50%. Whereas in LBP-DDF technique we improved further 4 body. So, total 10 body parts scored over 50% in LBP-DDF technique. In both techniques none of the body parts scored over 60%. However, in HOG out of 27 body parts we noticed that total 13 body parts have over 50% accuracy in which 8 body parts

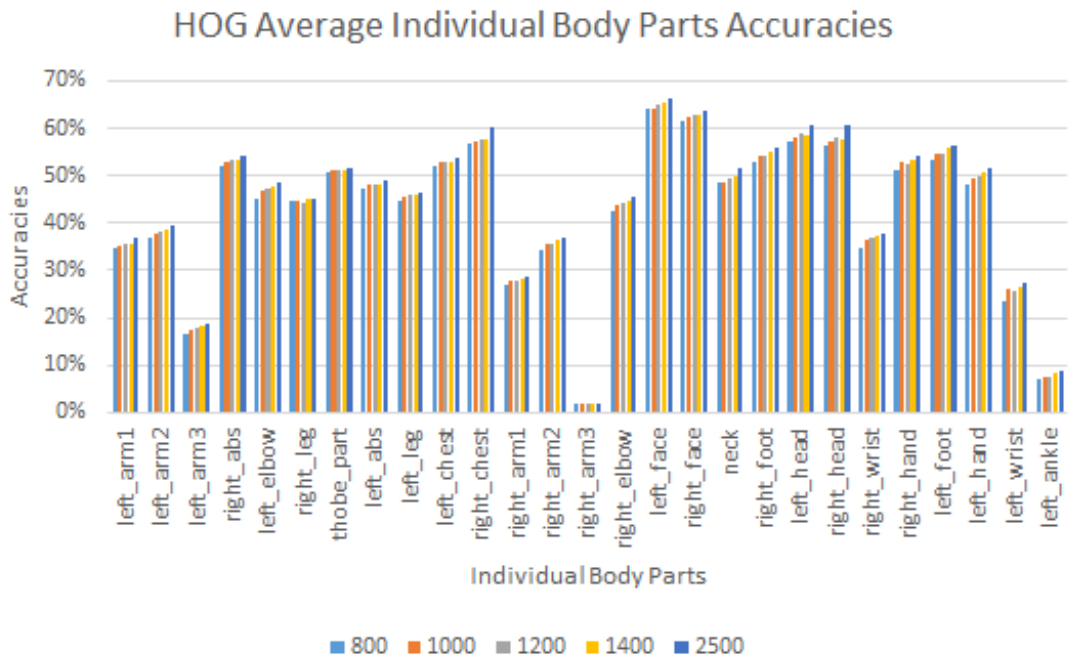


Figure 4.14: HOG Individual Body Parts Results

have over 50% accuracy and 5 body parts have over 60% accuracy. These statistics shows HOG produced better individual body parts results and better than depth techniques. First time we improved accuracy over 60% using HOG technique.

Upper and Lower Body Parts Results

Figure 4.15 shows the upper and lower body parts scores for different random number of pixels. These average classes accuracies are calculated from confusion matrix using equation 4.3. Both upper and lower body parts also scored below 50% in HOG technique. Surprisingly, In HOG we noticed that the lower body performed slightly less than upper body parts as compared to depth based features techniques. However, by looking at figure 4.16a and figure 4.16b it is clear that both upper and lower body parts are statistically significant. In others words both

Table 4.8: HOG Individual Body Parts Results

	left_arm1	**left_arm2**	**left_arm3**	**right_abs**	**left_elbow**	**right_leg**	**thobe_part**	**left_abs**	**left_leg**
800	34.55%	36.96%	16.54%	51.95%	45.15%	44.62%	50.75%	47.40%	44.89%
1000	35.17%	37.69%	17.64%	52.91%	46.86%	44.59%	51.08%	47.94%	45.66%
1200	35.73%	38.41%	18.05%	53.16%	47.22%	44.34%	51.12%	48.02%	45.79%
1400	35.78%	38.68%	18.50%	53.37%	47.91%	44.98%	51.29%	48.26%	45.94%
2500	**36.79%**	**39.50%**	**18.91%**	**54.15%**	**48.73%**	**45.10%**	**51.67%**	**48.92%**	**46.26%**
	left_chest	**right_chest**	**right_arm1**	**right_arm2**	**right_arm3**	**right_elbow**	**left_face**	**right_face**	**neck**
800	52.17%	56.71%	27.08%	34.42%	1.71%	42.68%	64.08%	61.62%	48.63%
1000	52.69%	57.14%	27.79%	35.55%	1.94%	43.92%	64.30%	62.43%	48.61%
1200	52.95%	57.56%	28.02%	35.50%	1.77%	44.13%	65.13%	62.93%	49.44%
1400	53.07%	57.68%	28.08%	36.30%	**2.00%**	44.76%	65.32%	62.92%	49.73%
2500	**53.78%**	**60.17%**	**28.74%**	**37.02%**	1.92%	**45.36%**	**66.21%**	**63.77%**	**51.41%**
	right_foot	**left_head**	**right_head**	**right_wrist**	**right_hand**	**left_foot**	**left_hand**	**left_wrist**	**left_ankle**
800	52.93%	57.03%	56.54%	34.72%	51.06%	53.25%	47.99%	23.44%	6.95%
1000	54.38%	58.00%	57.03%	36.37%	52.79%	54.64%	49.59%	26.00%	7.30%
1200	54.19%	59.03%	58.02%	36.74%	52.66%	54.72%	49.83%	25.76%	7.41%
1400	55.11%	58.54%	57.70%	37.25%	53.45%	55.71%	50.70%	26.73%	8.50%
2500	**55.76%**	**60.50%**	**60.51%**	**37.87%**	**54.03%**	**56.41%**	**51.67%**	**27.21%**	**8.86%**

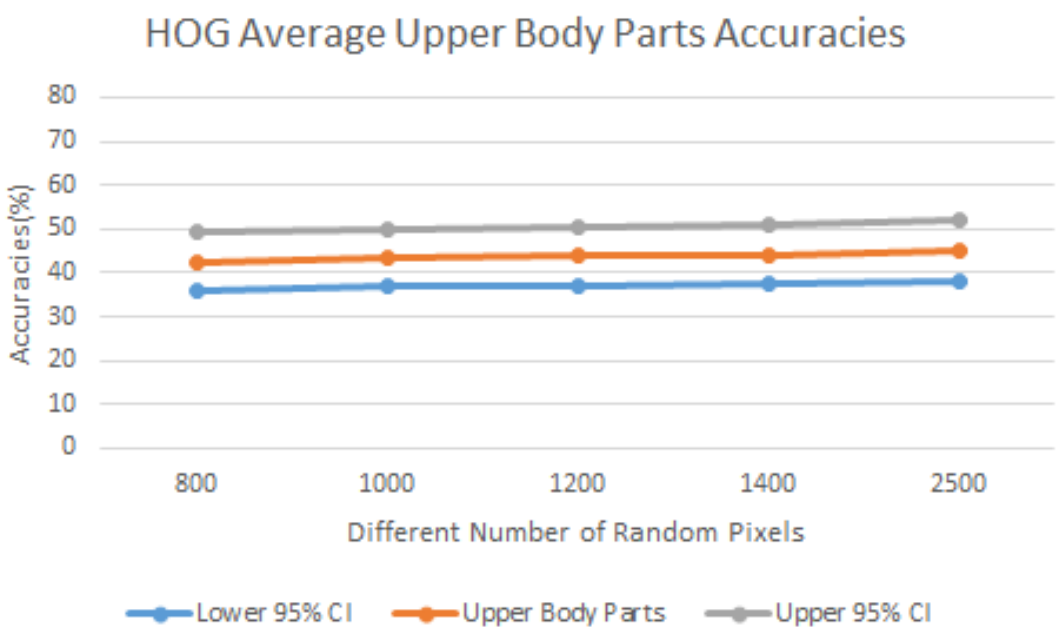
have equal accuracies.

We improved almost 3% accuracy for upper body parts from 800 random pixels to 2500 random pixels in figure 4.15a. The reason of having below 50% score in upper body parts is because of smaller body part e.g "left_arm3" and "right_arm3". These body parts were most of time occluded by a human pose and got lower number of random hits as mentioned in figure 4.46. That's why all these parts scored below 40%.

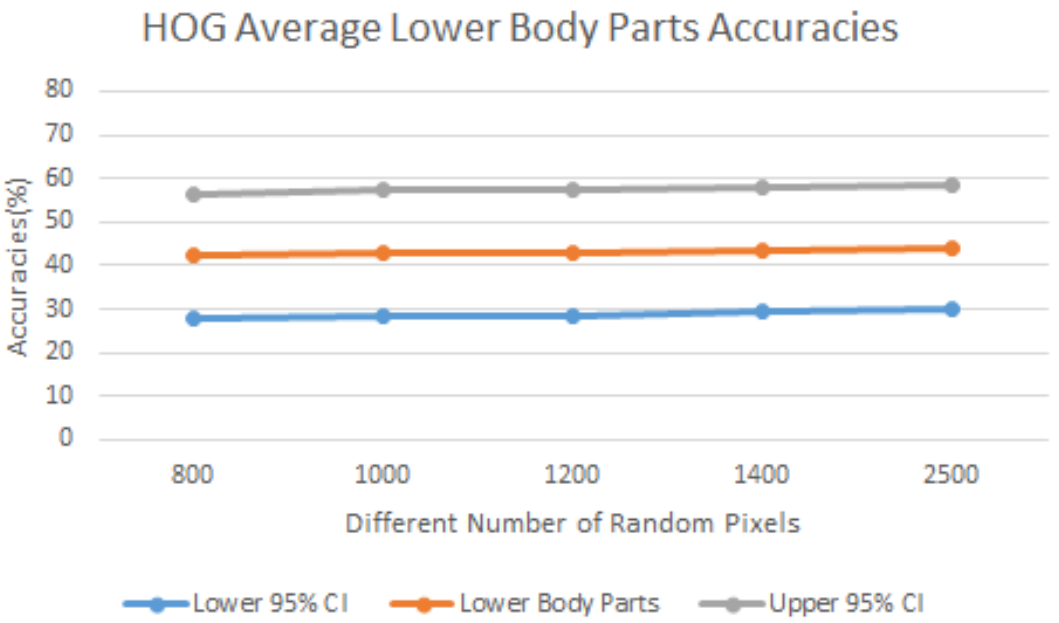
Similarly, The lower body parts also scored below 50%. Figure 4.15b shows that for every chunks of random pixels the average lower body parts decreased performance by 1% than average upper body parts score. However, the average lower body parts improved approximately 2% scored when we increased the number of random pixels from 800 to 2500. The highest accuracy we got is 44.01% at 2500 random pixels and lowest is 42.23% at 800 random pixels. All the lower body parts scored over 45% except the "left_ankle" body part which scored only 8.86%.

Figure 4.16 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.16a and figure 4.16b indicates that the mean of all the techniques are overlapping to each other in both upper and lower body parts. Therefore, we can conclude that we are 95% confident that the lower and upper body parts from 800 to 2500 random pixels in HOG technique are statistically significantly.

In comparison with depth based features technique. In HOG we improved both upper and lower body parts scores. In MDF technique we improved 5% score in



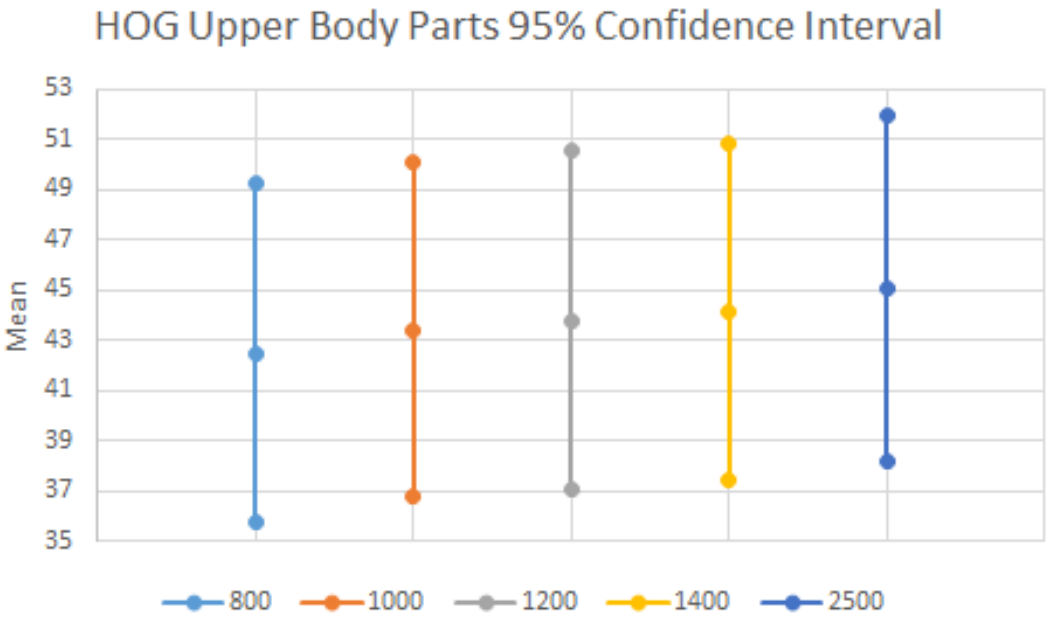
(a) HOG Average Upper Body Parts Accuracies



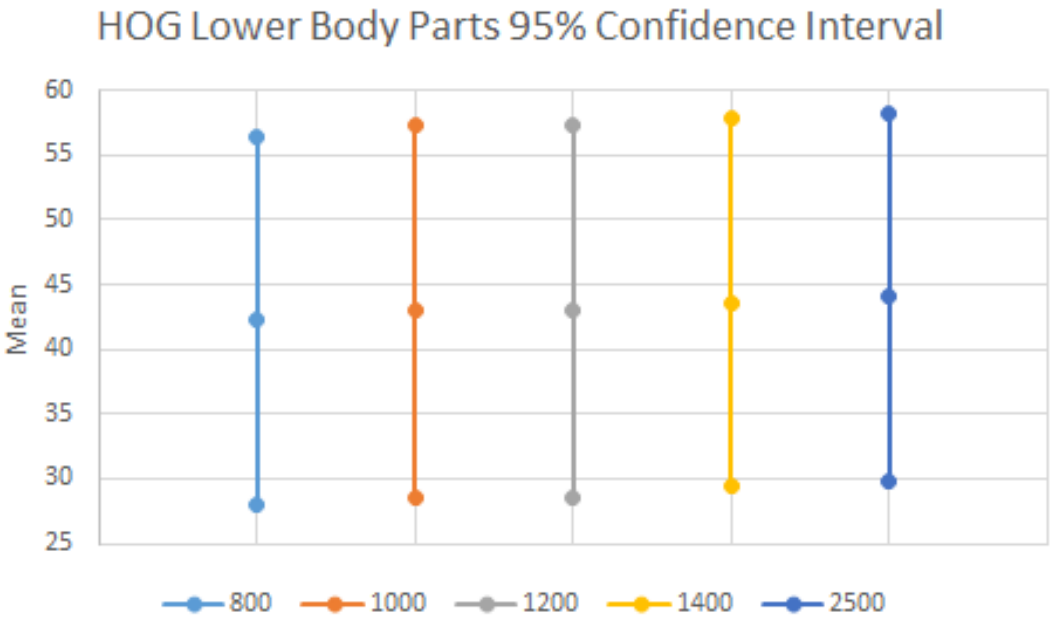
(b) HOG Average Lower Body Parts Accuracies

Figure 4.15: HOG Upper and Lower Body Parts Results

upper and lower body parts. Whereas in LBP-DDF technique we improved 2% score in upper body and 1% in lower body parts. Table 4.8 also indicated that we improved every single body part score from depth based features.



(a) HOG Upper Body Parts 95% Confidence Interval



(b) HOG Lower Body Parts 95% Confidence Interval

Figure 4.16: HOG Upper and Lower Body Parts Confidence Interval Results

4.4.4 HOG Average Results

In this technique we used HOG descriptor in a different way. We use the same window that we constructed in actual HOG technique and take the average of the window. We add the average value into the classified pixel x depth value and considered it a new depth feature.

However, by looking at the results of this technique in figure 4.17. Unfortunately, this technique does not performed well as expected. It performed worst than the original HOG technique. Even it performed worse than LBP-DDF technique. In comparison with MDF technique this technique produced little better score. The overall improvement in the accuracy is very low when we increased the random number of pixels. We got less than 1% improvement from 800 random pixels to 2500 random pixels.

Figure 4.18 shows the 95% confidence interval results. We noticed that all the random chunks of pixels have very low variance. It is because of the higher population size. In our case the population size we used is 30. We can say that we are 95% confident that the accuracy of each chunk of random pixel lies between the lower and upper confidence limit as mentioned in a figure 4.18.

However, if you look at the overlapping of ranges of different chunks of random pixels in figure 4.18. We figure it out that none of any random chunk of pixels intervals are overlapping. Therefore, we can say that the accuracies of all the random chunks pixels are statistically not significant. In other words every time we run the experiments all the chunks of random pixels will get the different

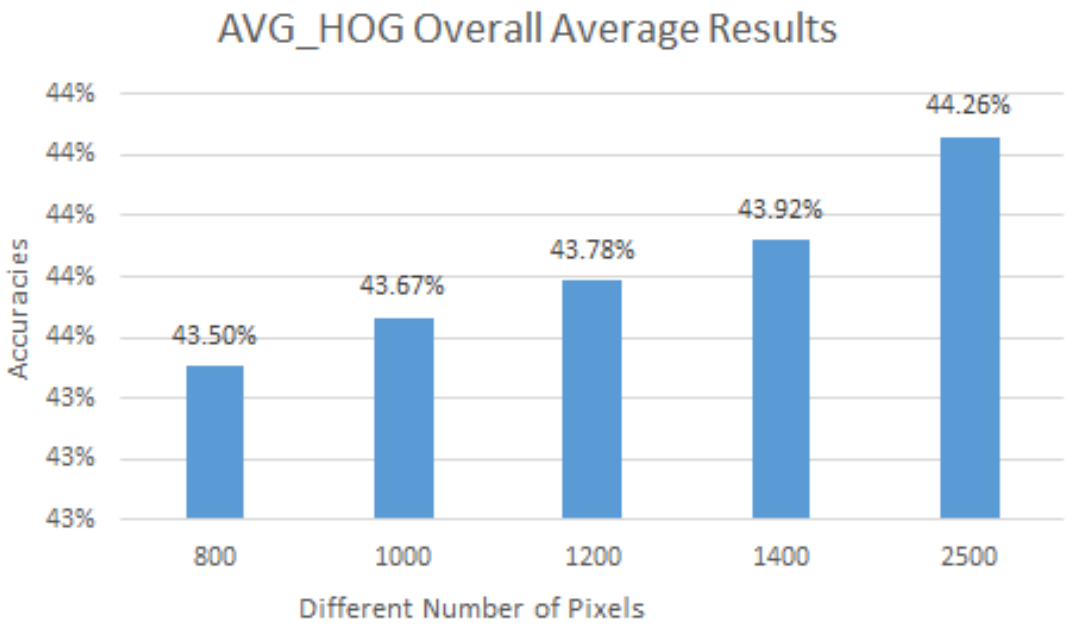


Figure 4.17: HOG Average Overall Results

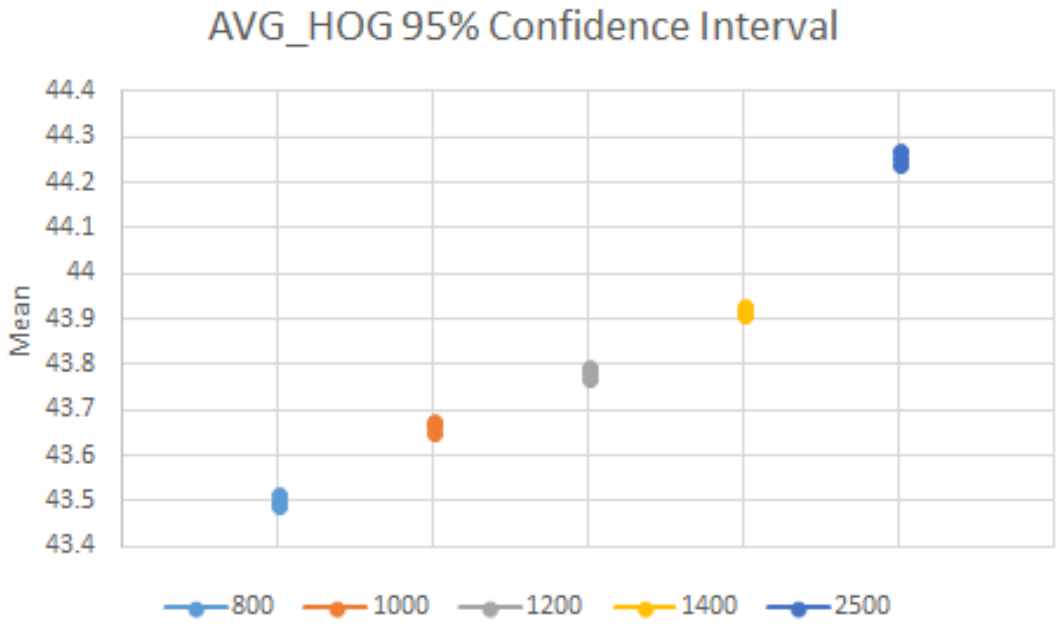


Figure 4.18: HOG Average 95% Confidence Interval

accuracies.

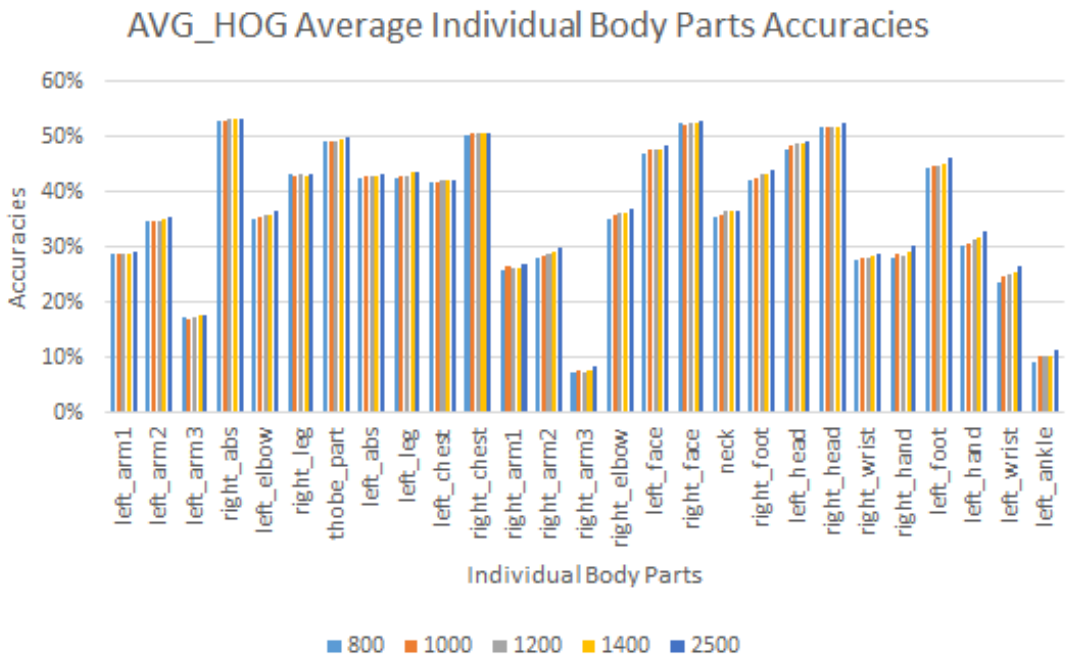


Figure 4.19: HOG Average Individual Body Parts Results

Individual Body Parts

The accuracies of individual parts are not satisfactory as compared to HOG and depth techniques. In table 4.9 almost all the body parts scored less than other techniques. Out of 27 body parts only 5 body parts have scored over 50%. None of the body part scored over 60%. The highest accuracy is 53.28% for the "right_abs" body part and lowest is 8.18% for "right_arm3" body part. However, in this technique the "right_arm3" body part is also performed lowest score like other previous techniques. Figure 4.19 shows the graphical representation of each body part accuracy using HOG average depth technique.

Table 4.9: HOG Average Individual Body Parts Results

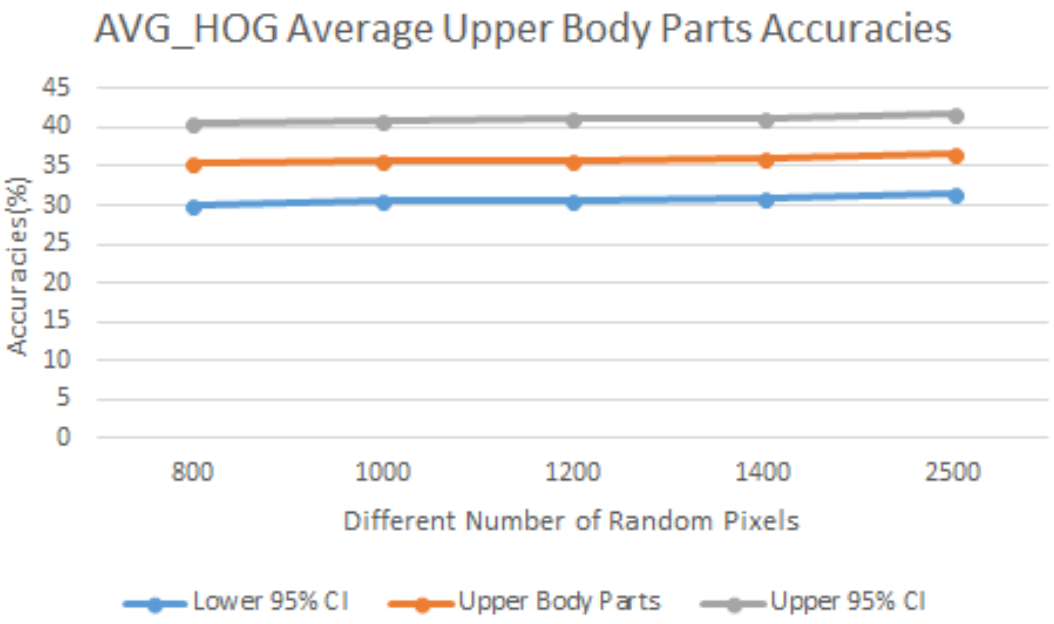
	left_arm1	**left_arm2**	**left_arm3**	**right_abs**	**left_elbow**	**right_leg**	**thobe_part**	**left_abs**	**left_leg**
800	28.60%	34.50%	17.32%	52.87%	34.94%	43.06%	49.30%	42.62%	42.60%
1000	28.69%	34.64%	17.03%	52.96%	35.45%	42.98%	49.20%	42.74%	42.97%
1200	28.88%	34.80%	17.39%	53.16%	35.69%	43.18%	49.20%	42.66%	42.98%
1400	28.86%	35.03%	**17.61%**	53.21%	35.69%	42.93%	49.42%	42.72%	**43.48%**
2500	**29.23%**	**35.32%**	17.58%	**53.28%**	**36.68%**	**43.37%**	**50.03%**	**43.19%**	43.39%
	left_chest	**right_chest**	**right_arm1**	**right_arm2**	**right_arm3**	**right_elbow**	**left_face**	**right_face**	**neck**
800	41.64%	50.36%	25.70%	28.09%	7.16%	34.96%	47.02%	52.38%	35.47%
1000	41.81%	50.52%	26.33%	28.38%	7.60%	35.65%	47.76%	52.20%	35.90%
1200	41.99%	50.53%	26.11%	28.80%	7.18%	35.99%	47.78%	52.41%	36.34%
1400	42.12%	50.62%	26.31%	28.97%	7.61%	36.33%	47.69%	52.60%	36.40%
2500	**42.25%**	**50.73%**	**26.97%**	**29.94%**	**8.18%**	**36.83%**	**48.22%**	**52.79%**	**36.68%**
	right_foot	**left_head**	**right_head**	**right_wrist**	**right_hand**	**left_foot**	**left_hand**	**left_wrist**	**left_ankle**
800	41.93%	47.63%	51.75%	27.76%	28.08%	44.39%	30.28%	23.70%	9.06%
1000	42.48%	48.48%	51.58%	27.98%	28.73%	44.54%	30.53%	24.83%	10.23%
1200	43.01%	48.90%	51.78%	28.16%	28.52%	44.74%	31.30%	24.97%	10.12%
1400	43.12%	48.84%	51.80%	28.29%	29.11%	45.08%	31.72%	25.38%	10.22%
2500	**43.97%**	**49.29%**	**52.30%**	**28.65%**	**30.13%**	**46.19%**	**32.78%**	**26.37%**	**11.24%**

Upper and Lower Body Parts Results

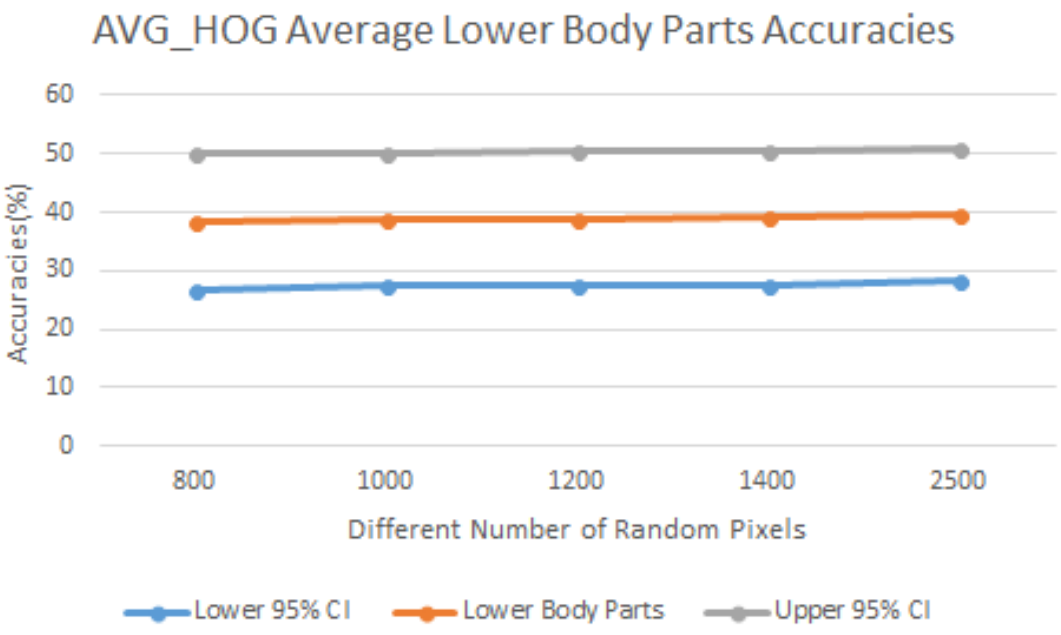
Figure 4.20 shows the upper and lower body parts scores for different random number of pixels. These average classes accuracies are calculated from confusion matrix using equation 4.3. Both upper and lower body parts also scored below up to 40%. In comparison with previous techniques. The HOG-AVG technique only improved 1% from MDF techniques.The rest of the techniques have better average upper and lower body parts accuracy than HOG-AVG.

Likewise, other previous techniques this technique also produced better average lower body parts score than upper body parts except HOG technique where average upper body parts scored better. However, when we increased the random number of pixels the increase in accuracy is very little. The upper body part increased almost 1.5% from 800 random pixels to 2500 random pixel in 4.20a. The lower body parts increased less than 3% from 800 random pixels to 2500 random pixel in 4.20b.

Figure 4.21 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.21a and figure 4.21b indicates that the mean of all the techniques are overlapping to each other in both upper and lower body parts. Therefore, we can conclude that we are 95% confident that the lower and upper body parts from 800 to 2500 random pixels are statistically significantly.

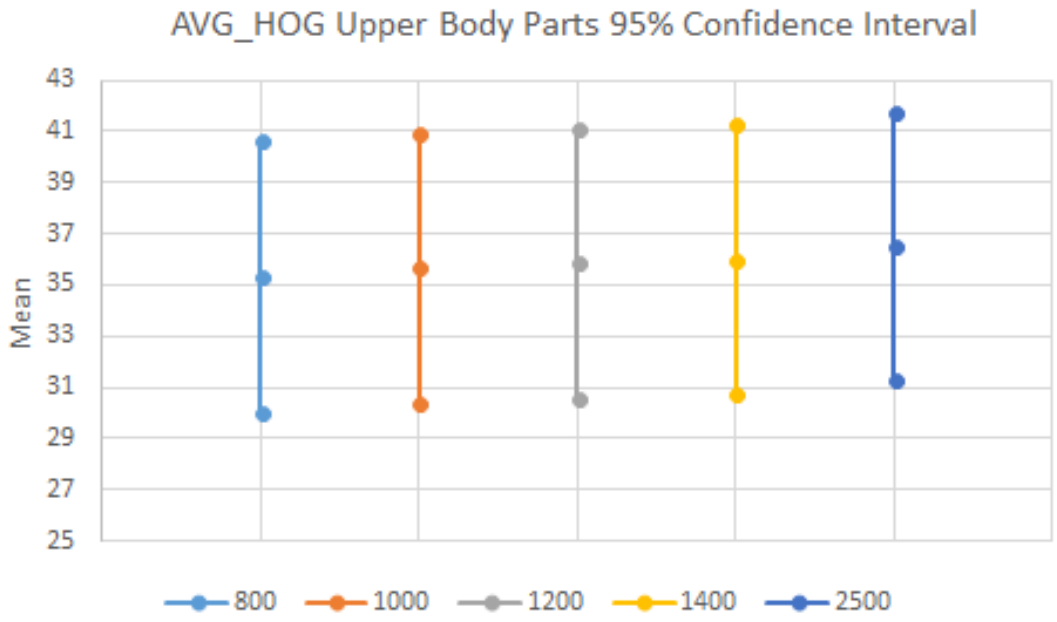


(a) HOG-AVG Average Upper Body Parts Accuracies

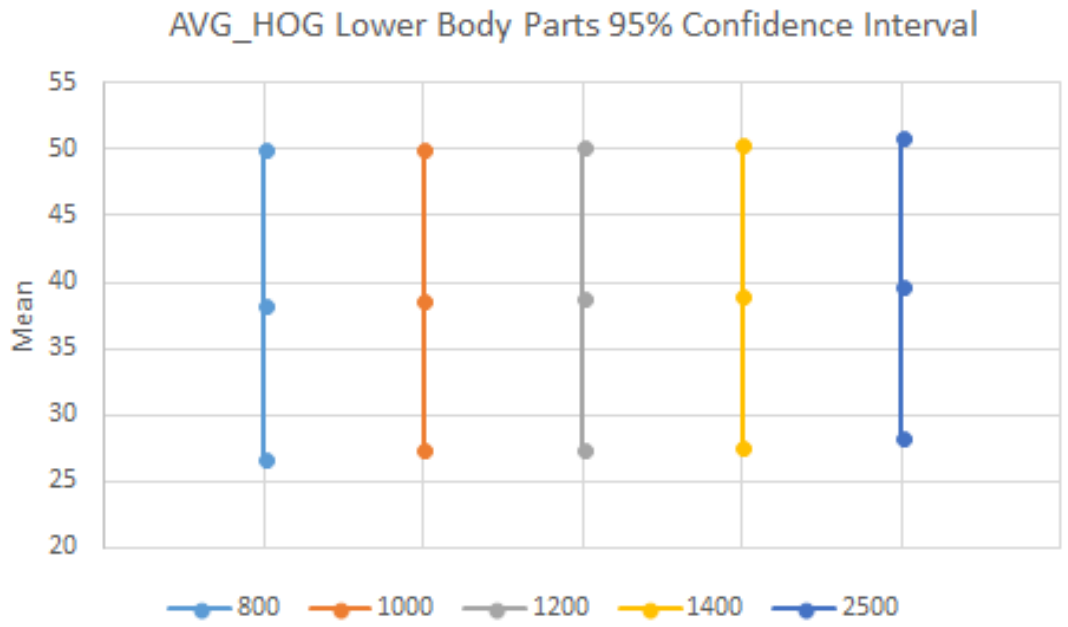


(b) HOG-AVG Average Lower Body Parts Accuracies

Figure 4.20: HOG-AVG Upper and Lower Body Parts Results



(a) HOG-AVG Upper Body Parts 95% Confidence Interval



(b) HOG-AVG Lower Body Parts 95% Confidence Interval

Figure 4.21: HOG-AVG Upper and Lower Body Parts Confidence Interval Results

4.4.5 SIFT Results

We used SIFT descriptor on different random pixels in depth image like HOG descriptor. SIFT descriptor is capable of capturing the local attributes/features of a image or an object. We used the VLFeat [79] library to implement SIFT descriptor in our thesis. VLFeat[79] is open source cross platform that provides bundle of popular computer vision algorithms libraries. The library for SIFT descriptor is robust and easy to implement in Matlab[80] environment. Like HOG in SIFT When a window is out of bound of the image it produced null vectors. We removed the null vectors from the feature vector during the classification phase.

SIFT Results were classified only for 800 random pixels. The rest of the different random pixels were computed but never classified. The size of the feature vector is too large to classified and takes lot of time. However, we recorded more than 10.6 million instances for training and more than 60 million instances for testing. We would not have enough processing power to classify the huge feature vector. Therefore, we were only able to classify SIFT descriptor only for 800 random pixels.

Figure 4.22 shows the average results of 30 runs for SIFT descriptor. We got very impressive accuracy which is much better than any other pixel based methods. We got 64.05% accuracy which is more than 18% from depth based techniques. Out of 100% instances the SIFT descriptor is able to detect 64% instances correctly. However, 37% of instances were incorrectly classified. The results can further be improved if we increased the number of random pixels and

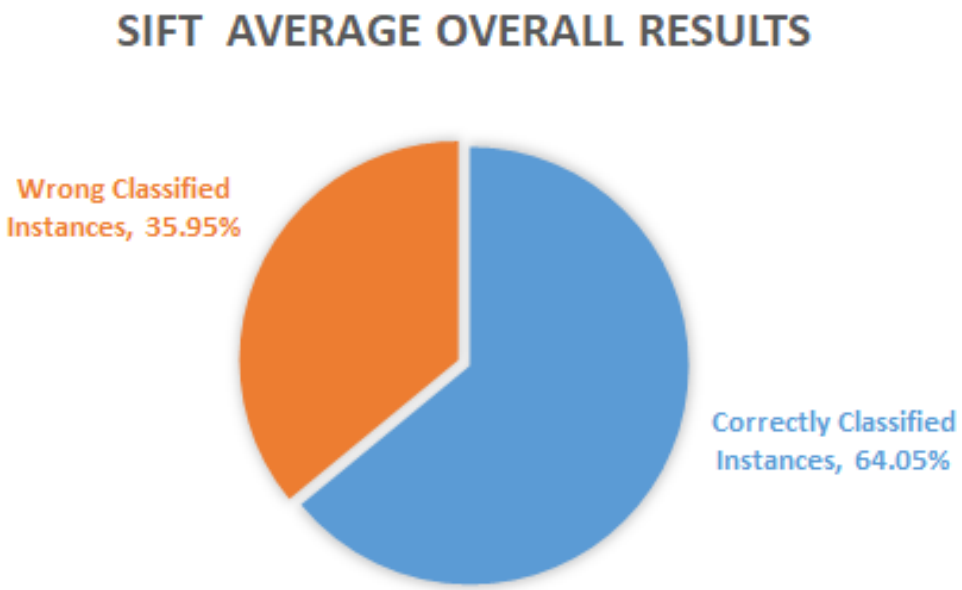


Figure 4.22: SIFT Overall Results

Table 4.10: SIFT Average 95% Confidence Interval

SIFT 95% Confidence Interval	
Mean	64.05215961
Standard Deviation (S.D)	0.952428578
Sample Size	30
Significance level (Alpha)	0.05
Lower 95% CI	63.7113343
Upper 95% CI	64.3929657

random decision trees. Table 4.10 shows the Statistical data for SIFT such as lower and upper confidence interval limit. The confidence interval in table 4.10 shows that we are 95% confident that the accuracy of SIFT lies in an upper and lower confidence interval limit.

Individual Body Parts

Table 4.11 represents the accuracies of 27 body parts using SIFT features technique. Similar to other pixel based techniques in this technique we also ran experiments up to 30 times and table 4.11 indicates the average accuracies of each body part.

In SIFT individual body parts we found very interesting results as compared all other pixel based techniques. In table 4.11 we recorded a notable accuracy improvements in almost every body parts. The highest performance jump were noted in "left_arm1", "left_chest", "left_ankle" and "right_arm1" body parts where we gained more than 20% accuracy as compared HOG technique. The "left_face" body part is leading with highest score which is 80.66% followed by "right_face". Similarly, like other previous techniques in SIFT the body parts "right_arm3" and "left_ankle" also performed worst among all other body parts. However, the accuracies of these body parts improved in SIFT by more than 15% as compared to all other techniques but still scored under 40%.

Out of 27 body parts only 5 body parts scored below 50% , 5 body parts scored over 50% and 17 body parts scored over 60% which is much better than HOG and any other techniques. Therefore, we can say that the SIFT performed very well in depth images specially on pixel based approach. Figure 4.23 shows the graphical representation of each body part accuracy using SIFT technique.

Table 4.11: SIFT Average Individual Body Parts Results

	left_arm2	**left_arm3**	**left_elbow**	**thobe_part**	**right_abs**	**left_arm1**	**right_leg**	**left_abs**	**left_leg**
800	60.09%	37.79%	60.02%	62.99%	67.92%	58.66%	56.71%	64.88%	60.11%
	left_chest	**right_chest**	**right_arm1**	**right_arm2**	**right_arm3**	**right_elbow**	**left_face**	**right_face**	**neck**
800	73.14%	74.53%	60.06%	60.74%	15.15%	57.58%	80.66%	80.07%	60.88%
	right_foot	**left_head**	**right_head**	**right_wrist**	**right_hand**	**left_foot**	**left_hand**	**left_wrist**	**left_ankle**
800	63.69%	78.10%	79.10%	50.16%	50.89%	63.78%	47.60%	41.78%	36.15%

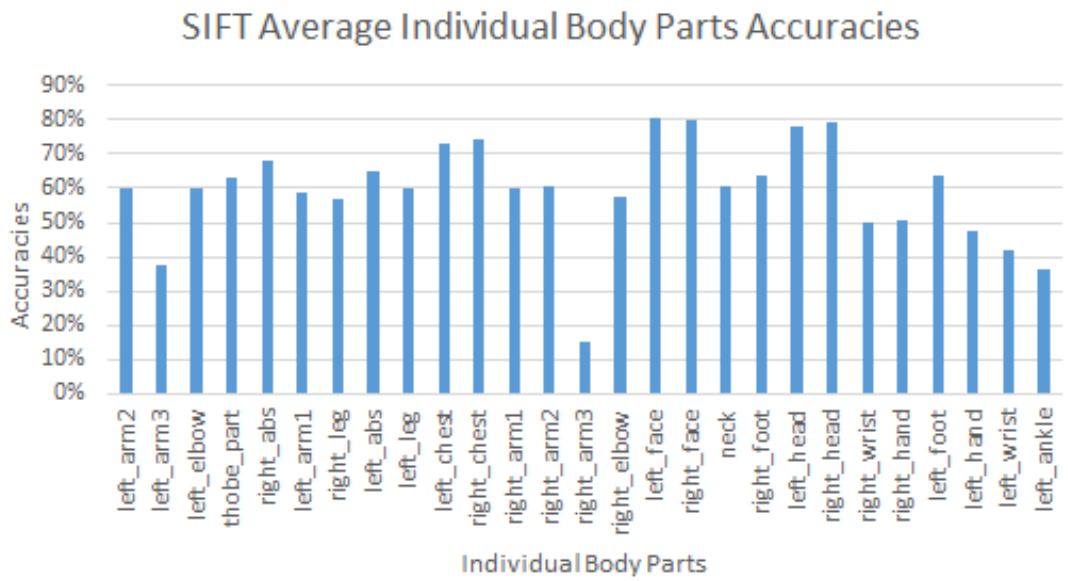


Figure 4.23: SIFT Individual body parts accuracies

Upper and Lower Body Parts Results

Figure 4.24 shows the upper and lower body parts scores for only 800 random number of pixels. These average classes accuracies are calculated from confusion matrix using equation 4.3. Both upper and lower body parts also scored almost 60% in SIFT technique. In SIFT we noticed that the lower body performed slightly less than upper body parts similar to HOG features techniques. However, by looking at figure 4.25 it is clear that both upper and lower body parts are statistically significant. In others words both have more or less equal accuracies.

Figure 4.24 shows the average results of upper and lower body parts. One thing we observed in all pixel based techniques that all have higher lower body parts accuracy than upper body parts except HOG. In SIFT the situation is similar to HOG. Table 4.11 states that all the lower body parts have more than 60% accuracy except "left_ankle" body part. The "right_foot" and "left_foot" scored

over 64%. The reason of having low lower body parts accuracy than upper body parts is because of "left_ankle" body part which performed less than 40%. This body part body parts got lower number of random hits as mentioned in figure 4.46. Whereas, the upper body parts scored almost 60%. However, there are few parts that have less than 50% accuracies such as "left_arm3", "right_ arm3", "left_hand" and "left_wrist". These body parts are very difficult to recognized correctly because of the smaller in size and occlusion. Therefore, in conclusion to SIFT we can say that SIFT produced better upper and lower body results as compared to other pixel based methods.

Figure 4.25 shows the 95% confidence interval graph for upper and lower body parts. Both upper and lower body mean values are overlapping to each other. Therefore, we can conclude that we are 95% confident that the lower and upper body parts from 800 random pixels in SIFT are statistically significantly.

In comparison with previous features technique. In SIFT we improved both upper and lower body parts scores. We improved more than 17% score in both upper and lower body parts.

4.4.6 Fusion of Features Based Results

We fused all the features together in order to check whether we can improve the accuracy or not. We know that all features have different characteristics that's why we fused depth feature with other depth feature and also fused depth with computer vision features techniques. Figure 4.26 shows the results of fusion

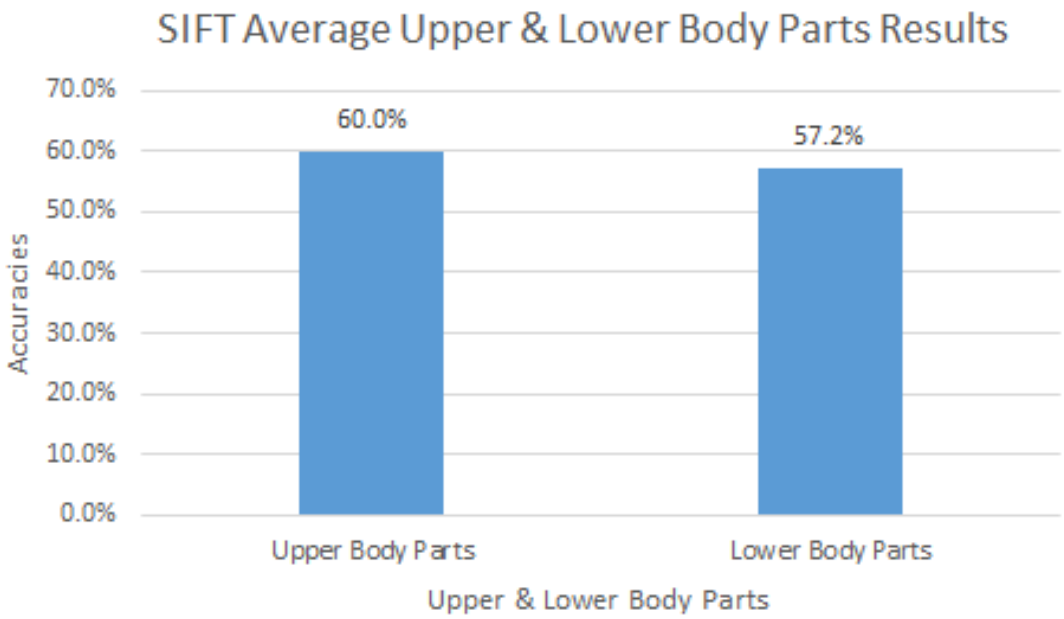


Figure 4.24: SIFT Upper and Lower body parts accuracies

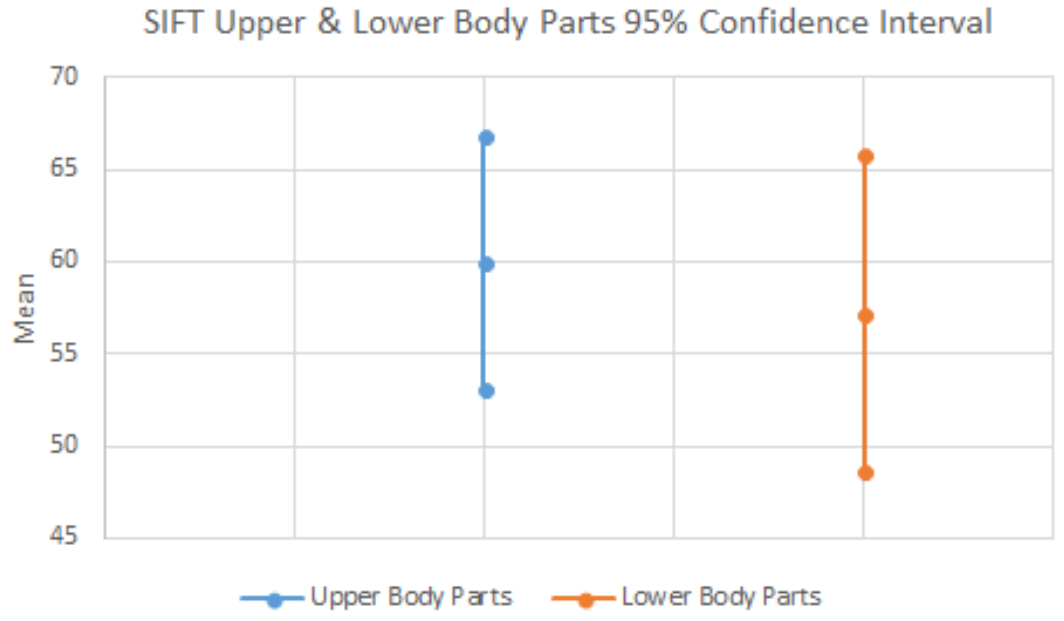


Figure 4.25: SIFT Upper and Lower body parts 95% Confidence Interval

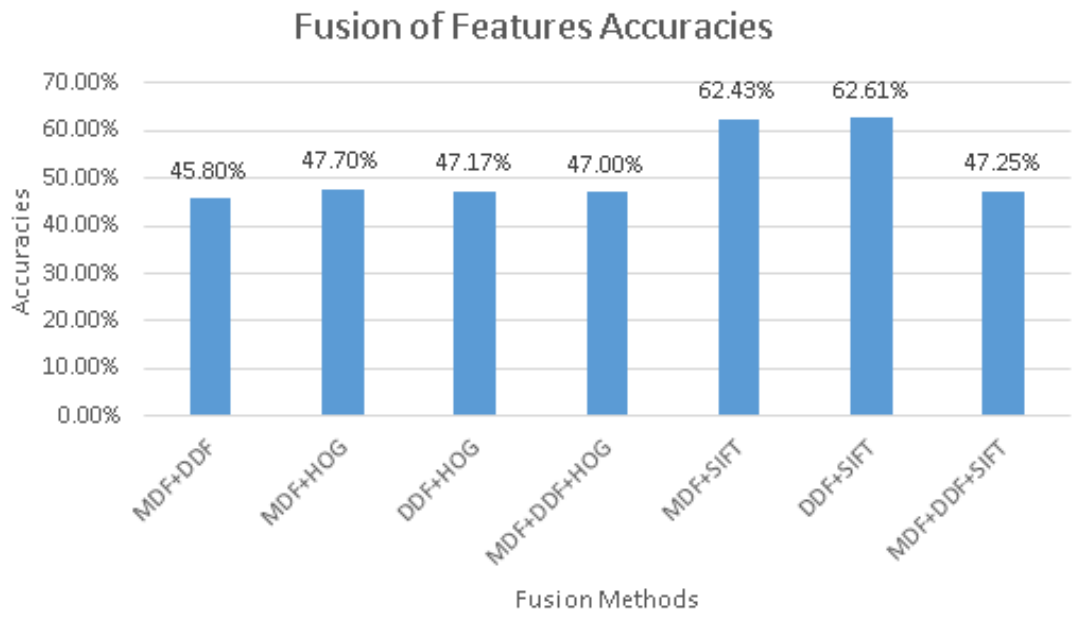


Figure 4.26: Fusion of Features Accuracies

features.

We observed from figure 4.26 that fusing features together never worked. We noticed a slight decline in the accuracies. The decimal depth feature alone gave almost 48% accuracy and when we combined the median depth feature with it. We lost 2% of accuracy. Similarly, when median and decimal depth feature is combined with HOG and SIFT the accuracy also fall down by 2% from HOG and SIFT actual accuracy. However, when we used both median and decimal depth features together with computer vision feature techniques(HOG and SIFT). We also noticed a decline in the accuracies. Interestingly, when we combined both depth feature with SIFT the accuracy is badly affected. The accuracy of the SIFT goes down to 47.25% which is almost 15% declined.

Therefore, we can conclude that mixing of depth features with computer vision features may not improve that accuracy.

Table 4.12: Patched Based Configurations

Configurations	
Total images	19000
No of patches in one image	26
Depth	20
Training images	16000
Testing images	3000

4.5 Patch-Based Approach Results

In this section we represented patched based approach results. Instead of taking random pixels from whole human body. We divided the human body into patches or parts. In patch based approach we have total 26 body patches or parts. For every patch or part we applied different feature extraction techniques to extract features and then classified using Random Forest model. We used two different techniques here. The first one is Bag of Features and second one is local features. Table 4.12 shows the Training and Testing configuration for patched based approach.

4.5.1 Bag of Features Results

This is one of the patch based approach. This technique has been widely used for object detection and image classification. We used HOG and SIFT descriptor for extracting features from the image patch or part. The reason of using HOG and SIFT descriptor for having a good comparison between global and local features in patch based approaches. However, we already know that the SIFT and HOG performed better in pixel based approach. Therefore, we used these both descrip-

tor for Bag of feature technique as well. Using HOG and SIFT descriptors we generated discrete features vocabulary from the large number of training samples.

Whereas in testing given an image features are extracted and assigned to the nearest cluster in discrete vocabulary. Afterwords, a normalized histogram is generated from the testing set that represent the actual features. We used different numbers clusters starting from 56 clusters and than increased to 100, 150 and upto 200 clusters. We classified the training and testing data using random forest model. The random forest model consist of total 40 decision trees. The final output is the average of 40 decision trees.

4.5.2 Bag of Features using HOG

Table 4.13 shows the HOG results under different clusters and trees. We noticed that in each cluster when we increased the number of trees the accuracy also improved. Total 9% accuracy gained from tree 3 to tree 40 in almost every set of cluster except cluster 200 where we got 11% jump in accuracy. We also noticed that at cluster 200 the performance of HOG is slightly declined or remains same. whereas at tree 30 and tree 40 the performance is almost the same no change recorded. The highest performance we recorded in HOG is 53% at cluster 200 and at tree 40. The lowest scored is 41% which is at cluster 200 and at tree 3.

Table 4.13 showing the results of one execution. However, we know that the random forest model for 40 trees have maximum accuracies as compared other random forest models. Therefore, we decided to re execute the random forest

Table 4.13: Patch Based Approach HOG Results

Clusters	Tree 3	Tree 10	Tree 20	Tree 30	Tree 40
56	42%	47%	49%	50%	51%
100	43%	49%	51%	52%	52
150	42%	48%	50%	51%	51%
200	41%	48%	50%	52%	**53%**

model for 40 trees up to 30 times and take the average accuracies of different clusters. Figure 4.27 showing the average score of different clusters for random forest model that consists of 40 trees. We noticed that by taking an average of 30 runs the acuuracies of different clusters declined by 1% or 2% as compared to single run in table 4.13. Overall the last 3 clusters 100, 150 and 200 scored almost same which is 51%.

The purpose of re execute the experiments up to 30 times is not only to take an average score but also calculate the confidence interval. The reason of calculating the confidence interval is to make sure that the different clusters are statistically significant or not. Figure 4.28 showing the 95% confidence interval of clusters 56 to 200 for random forest model of 40 trees. We can observed from the figure 4.28 that all the clusters have very narrow range or variance and none of the clusters are overlapping. Therefore, we can say that we are 95% confident that from clusters 56 to 200 for random forest model of 40 trees are statistically not significant. It means 95% of times the cluster scored within upper and lower limit without overlapping with other clusters.

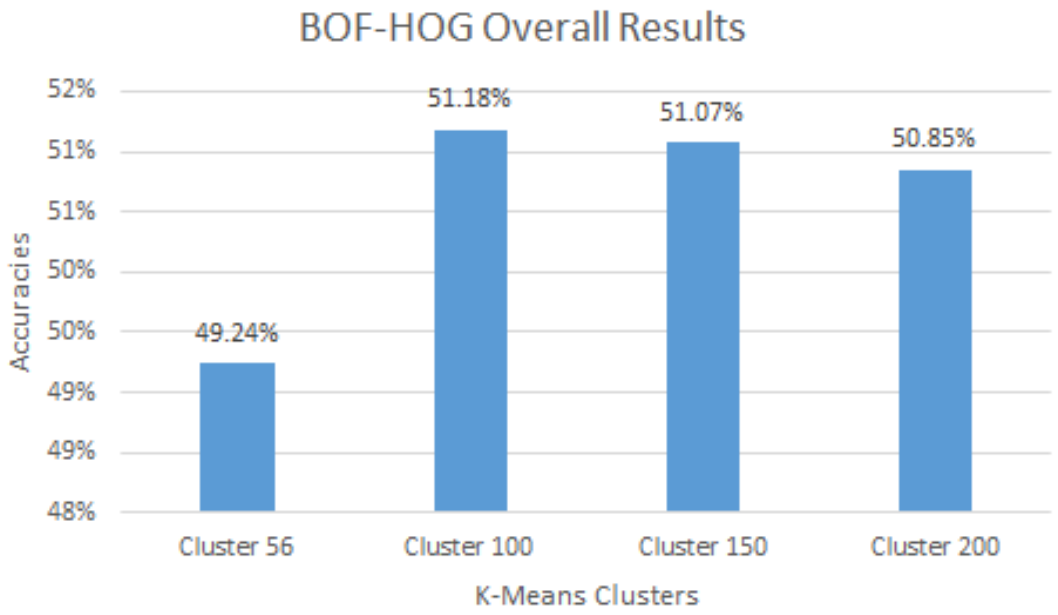


Figure 4.27: BOF-HOG Overall Results for Trees 40

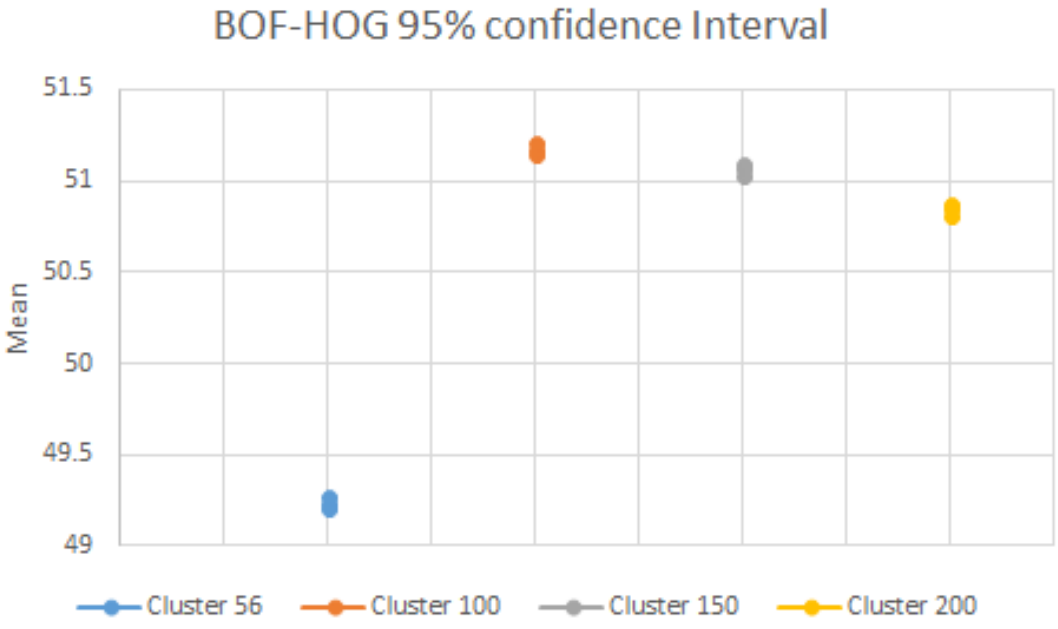


Figure 4.28: BOF-HOG Average 95% Confidence Interval for Trees 40

Individual Body Parts Results

The accuracies of individual body parts in patch based approach helps to figure out that how patches are correctly classified. We ran experiments up to 30 times and table 4.14 indicates the average accuracies of each body part of different clusters 40 trees random forest model. In table 4.14 "K" represents the number of K-Means clusters.

In this technique we have got mixed results among the different clusters unlike pixel based techniques. Where the highest number of random pixels chunks got the highest individual body parts accuracy. However, in this technique increasing the number of clusters decreased the individual body parts accuracy most of the time as described in table 4.14. Cluster 100 in table 4.14 predicted most of the body parts with better accuracy as compared to others cluster followed by cluster 56. The highest accuracy we got in table 4.14 is 84.03% at cluster 100 which is "left_chest". The lowest is 31.28% at cluster 150 which is "right_wrist".

The most important thing we have noticed in this patch based technique which is the improvements of smaller body parts accuracies like "right_arm3", and "left_arm3". These body parts performed very worst in pixel based techniques by having less than 10% accuracy. However, in this technique we improved almost 36% accuracy in "right_arm3" body part and more than 20% accuracy improved in "left_arm3" body parts. The reason behind the improvements because this technique is dealing with patches where the descriptor applied on all pixels rather than taking random pixels. Figure 4.29 shows the graphical representation

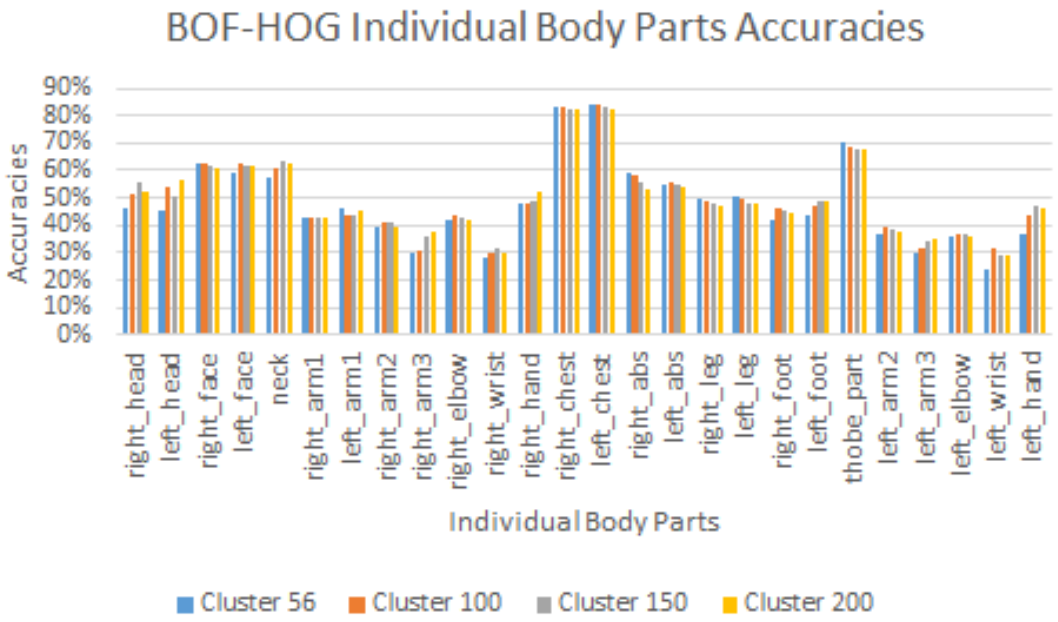


Figure 4.29: BOF-HOG Average Individual Body Parts Results for Trees 40

of each body part accuracy using BOF-HOG.

Upper and Lower Body Parts Results

Figure 4.30 shows the upper and lower body parts scores of different K-Means clusters. These average classes accuracies are calculated from confusion matrix using equation 4.3. The upper body parts scored approximately 50%. The lower body parts scored almost 52%.

We improved almost 2% accuracy for upper body parts from cluster 56 to 200 in figure 4.30a. We noticed at cluster 150 and 200 the upper body score is same. This is because when we increased number the clusters the performance of this technique start decreasing or remain same see table 4.13 after cluster 100. The lowest upper body accuracy we got at cluster 56 which is 47.25%. Similarly, The lower body parts scored over 50%. Figure 4.30b shows that the accuracy improved

Table 4.14: BOF-HOG Average Individual Body Parts Results for Trees 40

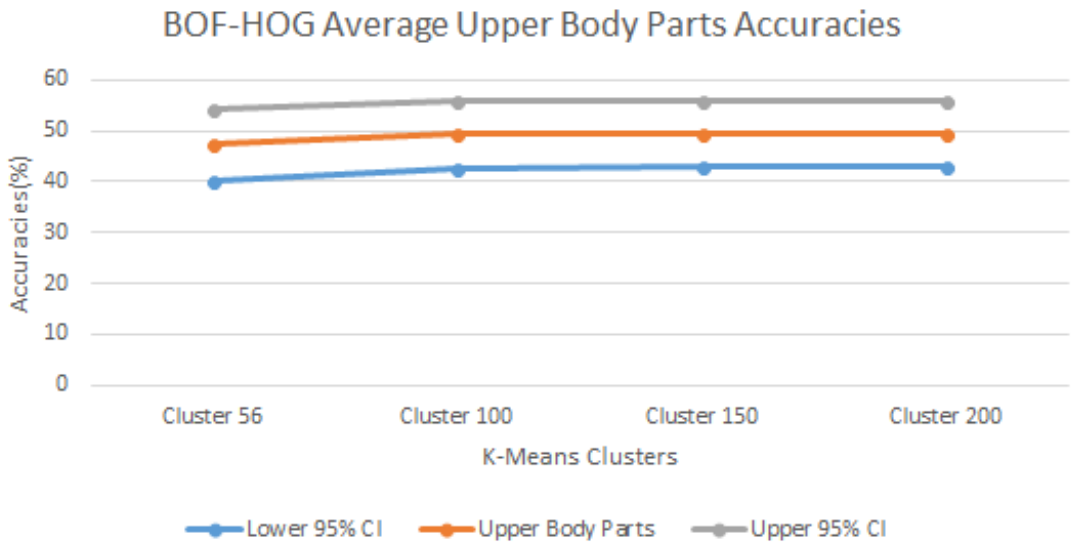
K	**right_head**	**left_head**	**right_face**	**left_face**	**neck**	**right_arm1**	**left_arm1**	**right_arm2**	**right_arm3**
56	46.37%	45.70%	62.37%	59.18%	57.48%	**43.12%**	**46.17%**	38.96%	29.70%
100	51.70%	53.99%	**62.52%**	**62.75%**	61.12%	42.71%	43.67%	40.64%	30.79%
150	**55.26%**	50.22%	61.82%	61.76%	**63.42%**	42.51%	43.28%	**40.76%**	35.94%
200	52.12%	**56.72%**	60.40%	61.52%	62.78%	42.52%	45.16%	39.11%	**38.00%**
K	**right_elbow**	**right_wrist**	**right_hand**	**right_chest**	**left_chest**	**right_abs**	**left_abs**	**right_leg**	**left_leg**
56	42.29%	28.41%	48.01%	82.97%	83.89%	**59.28%**	55.08%	**50.00%**	**50.51%**
100	**43.80%**	29.85%	48.00%	**83.42%**	**84.03%**	58.50%	**55.77%**	48.75%	49.42%
150	42.36%	**31.28%**	48.37%	82.65%	83.21%	55.28%	55.18%	47.69%	48.20%
200	41.69%	29.82%	**52.14%**	82.36%	82.73%	53.29%	54.32%	47.27%	48.05%
K	**right_foot**	**left_foot**	**thobe_part**	**left_arm2**	**left_arm3**	**left_elbow**	**left_wrist**	**left_hand**	
56	42.31%	43.50%	**70.04%**	36.76%	30.26%	35.70%	23.70%	36.89%	
100	**46.01%**	46.95%	68.64%	**39.26%**	31.82%	**37.16%**	**31.53%**	43.96%	
150	45.15%	48.97%	67.71%	38.36%	34.10%	36.59%	29.07%	**47.31%**	
200	44.06%	**49.08%**	67.52%	38.00%	**35.17%**	35.65%	28.92%	46.24%	

from cluster 56 to cluster 100. However, the lower body accuracy slightly decreased from cluster 100 to cluster 200.

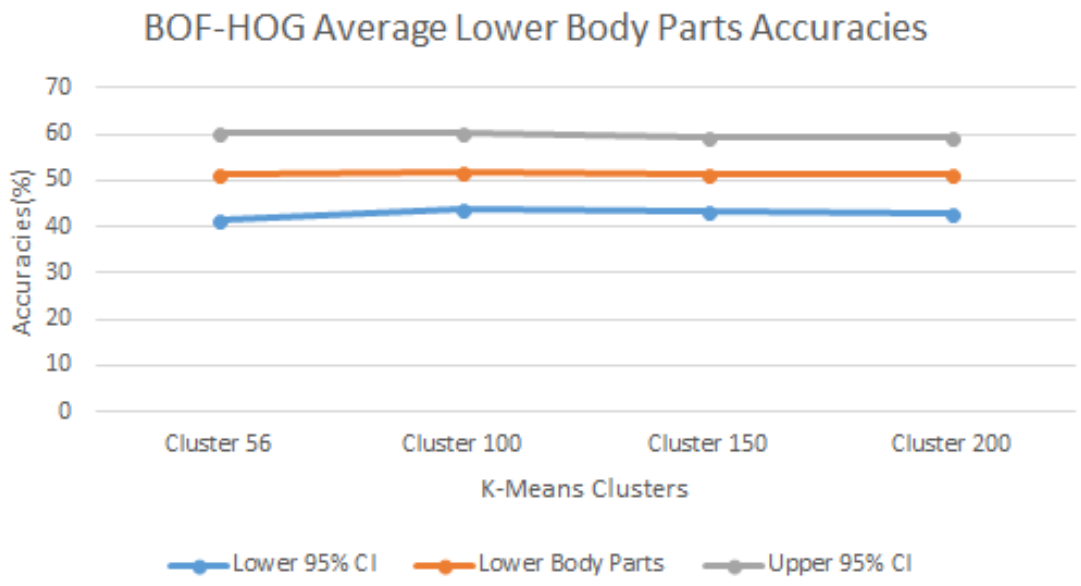
Figure 4.31 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.31a and figure 4.31b indicates that the mean of all the techniques are overlapping to each other in both upper and lower body parts. Therefore, we can conclude that we are 95% confident that the lower and upper body parts from cluster 56 to cluster 200 using random forest model of 40 trees are statistically significantly. It means their is no difference among the clusters all have same upper and lower body score. Cluster 100 and 150 have exactly same interval in figure 4.31a and figure 4.31b.

4.5.3 Bag of Features using SIFT

Table 4.15 shows patch based SIFT results using bag of feature technique. The results are not impressive for SIFT in patch based approach. The highest accuracy we got is 33% and lowest is 24%. However, there are two ways to interpret the results of BOF-SIFT in table 4.15. The first one when we increased the number of trees in the random forest model the performance is improved very little. We got only 1% accuracy jump from tree 3 to tree 40 which is less than BOF-HOG. As we can see that from figure 4.15 after at $10th$ tree the performance is remains the same. There is no significant change recorded. The second one when we increased the number of clusters. It has been noticed from table 4.15 that increase in number of clusters also increased the accuracy.

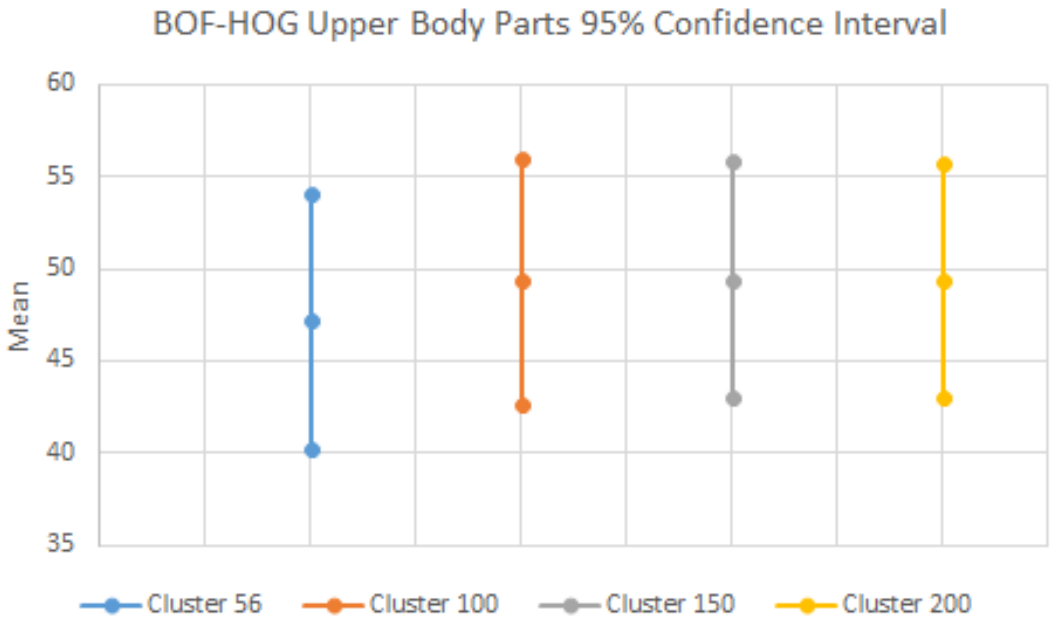


(a) BOF-HOG Average Upper Body Parts Accuracies for Trees 40

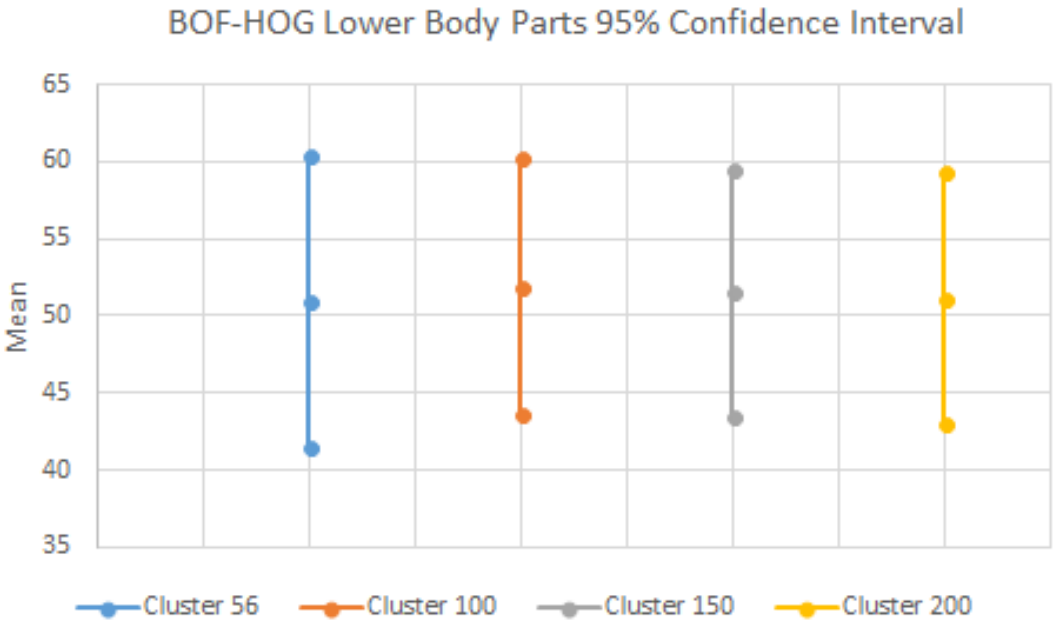


(b) BOF-HOG Average Lower Body Parts Accuracies for Trees 40

Figure 4.30: BOF-HOG Upper and Lower Body Parts Results



(a) BOF-HOG Upper Body Parts 95% Confidence Interval for Trees 40



(b) BOF-HOG Lower Body Parts 95% Confidence Interval for Trees 40

Figure 4.31: BOF-HOG Upper and Lower Body Parts Confidence Interval Results

Table 4.15 showing the results of one execution. Similar to BOF-HOG we decided to re execute the random forest model for 40 trees up to 30 times and take the average accuracies of different clusters. Figure 4.32 showing the average scores for different clusters using random forest model that consists of 40 trees. We noticed that by taking an average of 30 runs the accuracies of different clusters have similar response as in table 4.15. The accuracy increased as we increased the number of clusters. We got almost similar accuracies for 40 trees as recorded in table 4.15.

Figure 4.33 showing the 95% confidence interval of clusters 56 to 200 for random forest model of 40 trees. We can observed from the figure 4.33 that all the clusters have very narrow range or variance and none of the clusters are overlapping. Therefore, we can say that we are 95% confident that from clusters 56 to 200 for random forest model of 40 trees are statistically not significant. It means 95% of times the cluster scored within upper and lower limit without overlapping with other clusters. Every time the accuracy of each cluster is different than other clusters.

Table 4.15: Patch Based Approach sift Results

Clusters	**Tree 3**	**Tree 10**	**Tree 20**	**Tree 30**	**Tree 40**
56	24%	25%	25%	25%	25%
100	28%	29%	29%	29%	29%
150	30%	30%	31%	31%	31%
200	31%	31%	32%	32%	**33%**

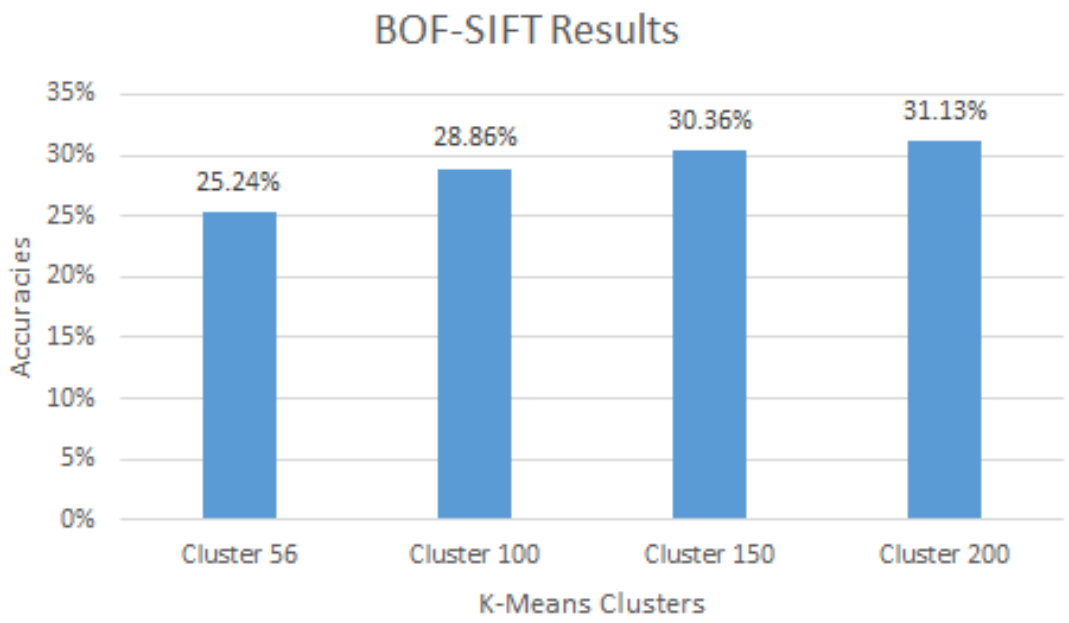


Figure 4.32: BOF-SIFT Overall Results for Trees 40

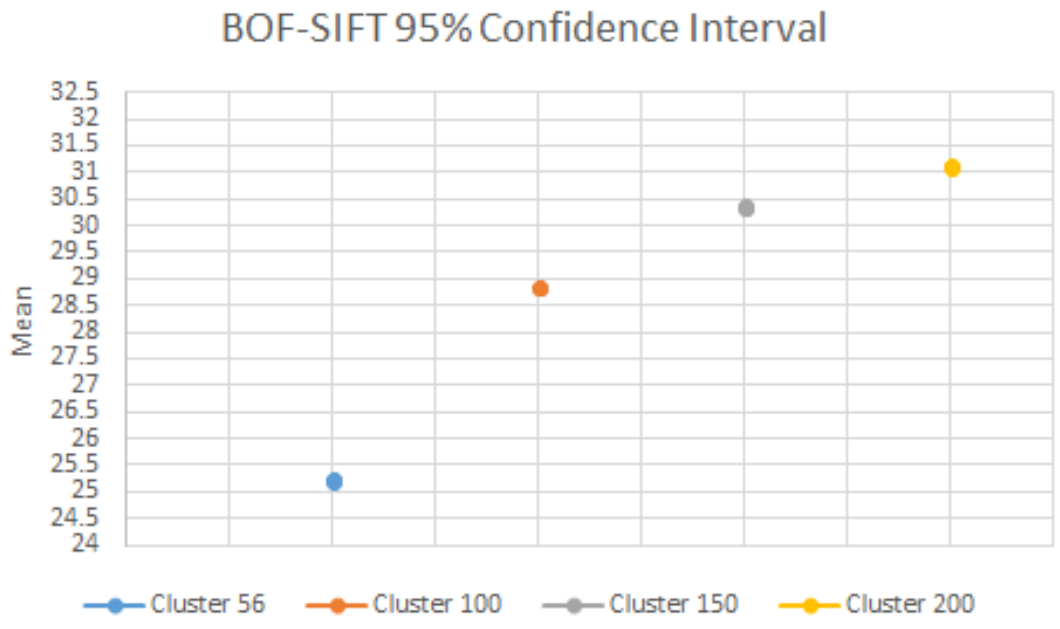


Figure 4.33: BOF-SIFT Average 95% Confidence Interval for Trees 40

Individual Body Parts Results

In this technique we ran experiments up to 30 times and table 4.16 indicates the average accuracies of each body part for different clusters of 40 trees random forest model. In table 4.16 "K" represents the number of K-Means clusters.

In this technique we have also got mixed results among the different clusters which is similar to BOF-HOG techniques. Cluster 150 in table 4.16 predicted most of the body parts with better accuracy as compared to others clusters followed by cluster 200. The highest accuracy we got in table 4.16 is 73.42% at cluster 200 which is "neck". The lowest is 14.99% at cluster 56 which is "right_arm3". The body parts "right_arm3" and "right_elbow" performed worst. Their accuracy is less than 20%.

In comparison with BOF-HOG this technique performed worst in almost every individual body parts. Out of 26 only 2 body parts scored over 50%. Which worst than all previous techniques including pixel based techniques. Figure 4.34 shows the graphical representation of each body part accuracy using BOF-SIFT.

Upper and Lower Body Parts Results

Figure 4.35 shows the upper and lower body parts scores of different K-Means clusters. These average classes accuracies are calculated from confusion matrix using equation 4.3. Both upper and lower body parts scored less than 50%. This techniques have lowest score than all previous techniques including pixel based techniques. The majority of the body parts scored less than 40% and this is the

Table 4.16: BOF-SIFT Average Individual Body Parts Results for Trees 40

K	right_head	left_head	right_face	left_face	neck	right_arm1	left_arm1	right_arm2	left_arm2
56	42.91%	20.09%	39.20%	36.96%	66.51%	13.15%	26.11%	13.90%	13.63%
100	45.56%	46.89%	41.46%	45.51%	68.95%	**33.10%**	28.52%	23.46%	9.70%
150	**47.71%**	**48.60%**	43.67%	**50.01%**	73.32%	29.98%	31.17%	**25.30%**	5.92%
200	46.32%	48.55%	**43.88%**	46.63%	**73.42%**	30.71%	**34.50%**	19.38%	**23.27%**
K	**right_arm3**	**left_arm3**	**right_elbow**	**left_elbow**	**right_wrist**	**right_hand**	**right_chest**	**left_chest**	**right_abs**
56	**14.99%**	12.05%	11.87%	12.67%	21.24%	16.77%	20.03%	**27.88%**	11.41%
100	12.77%	19.14%	8.22%	18.31%	**26.98%**	16.49%	25.04%	23.92%	12.46%
150	6.84%	**22.45%**	11.29%	**21.50%**	23.84%	23.38%	26.08%	24.75%	**22.52%**
200	10.41%	21.18%	**16.54%**	21.35%	26.27%	**25.12%**	**30.64%**	25.88%	20.14%
K	**left_abs**	**right_leg**	**left_leg**	**right_foot**	**left_foot**	**thobe_part**	**left_wrist**	**left_hand**	
56	22.22%	21.08%	19.60%	24.74%	24.91%	44.38%	17.50%	16.00%	
100	**26.46%**	21.25%	19.68%	27.25%	29.07%	43.36%	18.72%	15.85%	
150	24.30%	**22.11%**	19.81%	25.05%	**33.92%**	43.27%	**23.25%**	**23.51%**	
200	26.34%	21.90%	**20.26%**	**29.68%**	31.16%	**43.62%**	20.55%	21.75%	

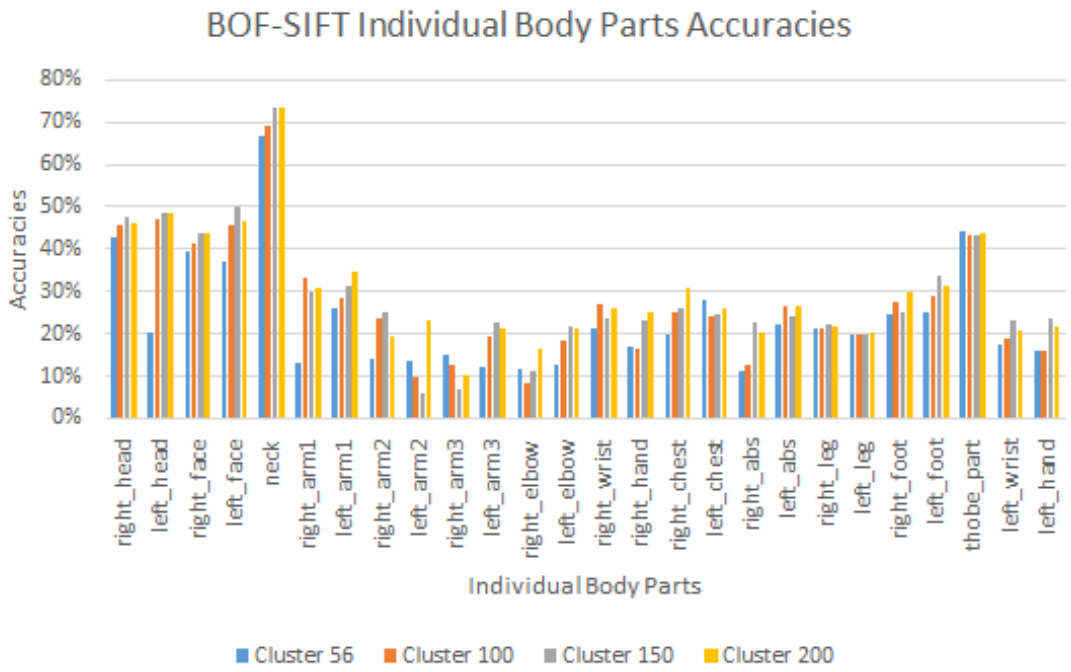


Figure 4.34: BOF-SIFT Average Individual Body Parts Results for Trees 40

reason of having very less score in upper and lower body parts.

However, by looking at upper body parts score in figure 4.35a. We improved almost 4% accuracy from cluster 56 to 200 in figure 4.35a. We noticed at cluster 150 and 200 the upper body scored higher than lower body. This is because the "neck" body part scored 70% in both clusters. The other reason is the upper body parts performed better after cluster 100. Cluster 150 and 200 improved 20 body parts than first two clusters. The lowest upper body accuracy we got at cluster 56 which is 27.72% and the highest is 30.13% at cluster 200. The lower body parts scored less than 30%. Figure 4.35b shows that very little accuracy improved from cluster 56 to cluster 100. Out of 5 lower body parts only 1 body parts scored over 40% and the rest scored less than 35%. This is worst ever lower body score we have been noticed so far.The lowest lower body accuracy we got at cluster 56

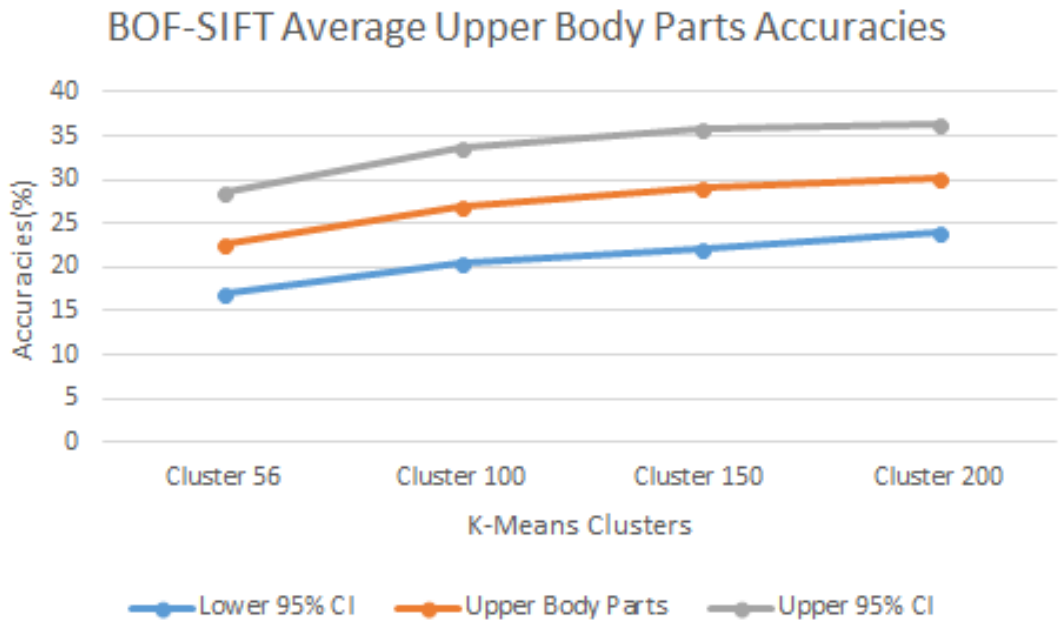
which is 26.94% and the highest is 29.32% at cluster 200.

Figure 4.36 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.36a and figure 4.36b indicates that the mean of all the techniques are overlapping to each other in both upper and lower body parts. Therefore, we can conclude that we are 95% confident that the lower and upper body parts from cluster 56 to cluster 100 using random forest model of 40 trees are statistically significantly. It means their is no difference among the clusters all have same upper and lower body score.

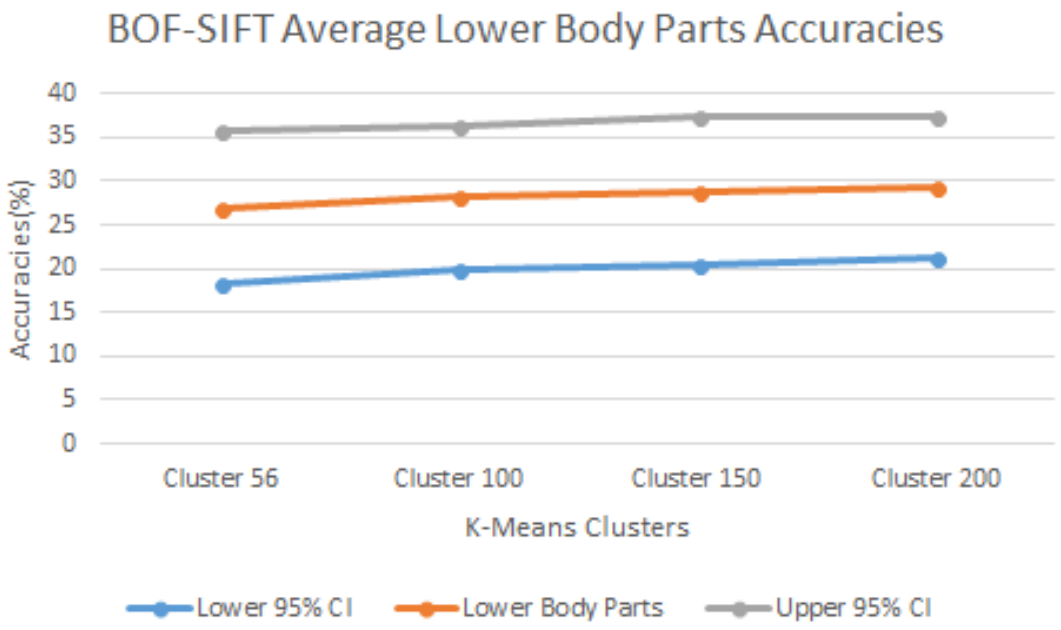
4.5.4 Local features Results

As discussed earlier the local features are used to find out the unique patterns or structures in an image that can be recognized as image patch, edge and point. Local features are used to play an important role in object detection specially when dealing with image patches. It represent the patch content so nicely for detection purposes. These features are very robust to noise, occlusion and change in viewing condition.

We used this technique in depth images where we extracted all the patches or body parts separately. By applying this on each patch we got some dominant and impotent features. To get rid of impotent features that are considered useless. We used Singular Value Decomposition (SVD) method to reduce the feature space. The SVD is robust and it use eigen values and eigen vectors to reduce the features. If the size of patch is nxm then the length of feature vector for each

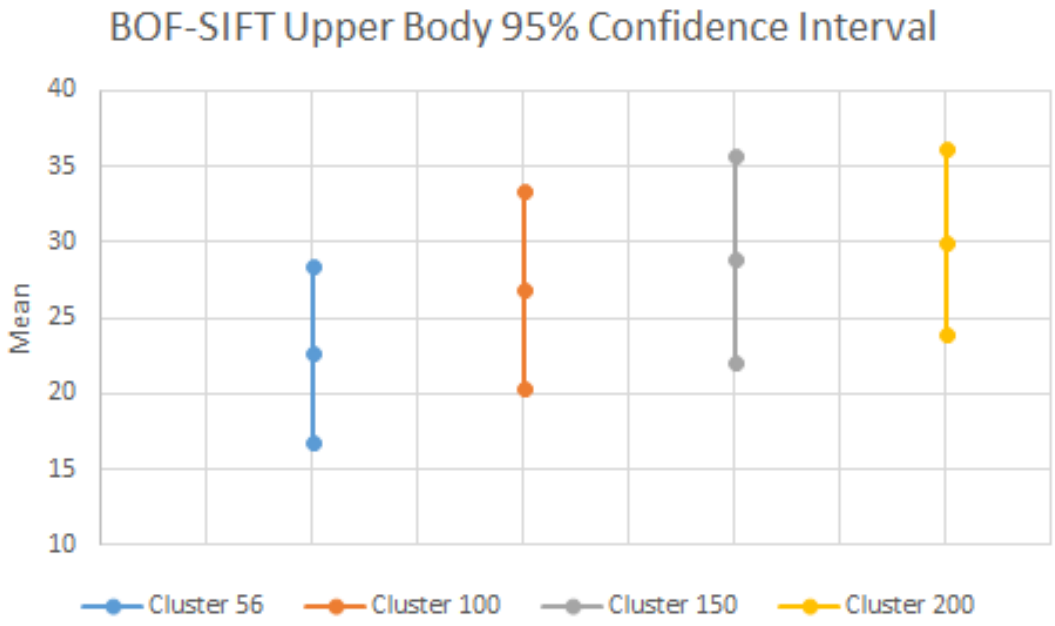


(a) BOF-SIFT Average Upper Body Parts Accuracies for Trees 40

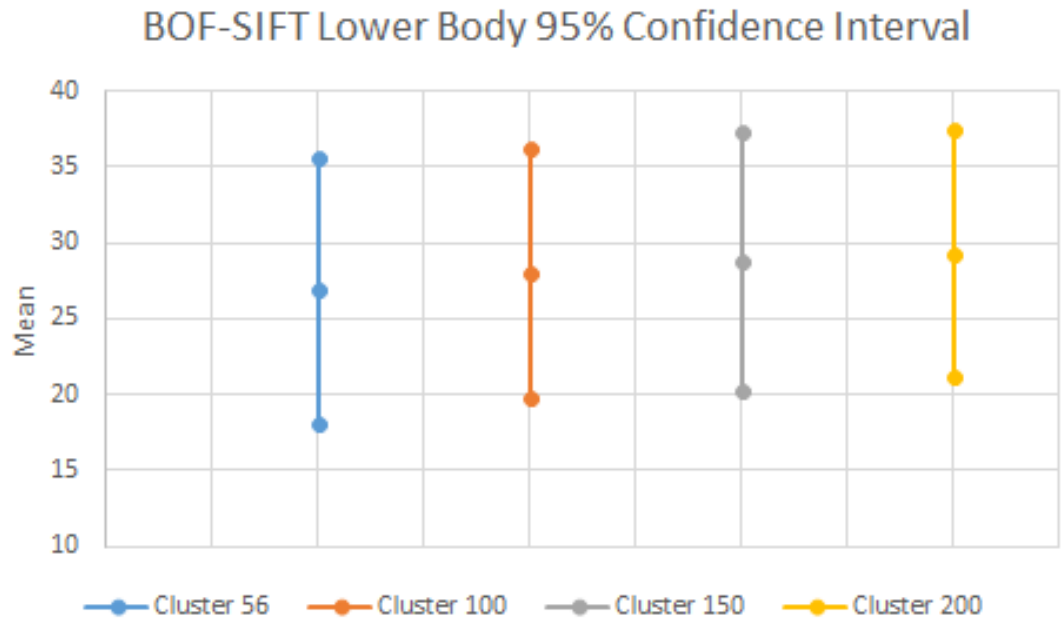


(b) BOF-SIFT Average Lower Body Parts Accuracies for Trees 40

Figure 4.35: BOF-SIFT Upper and Lower Body Parts Results



(a) BOF-SIFT Upper Body Parts 95% Confidence Interval for Trees 40



(b) BOF-SIFT Lower Body Parts 95% Confidence Interval for Trees 40

Figure 4.36: BOF-SIFT Upper and Lower Body Parts Confidence Interval Results

patch is $2m$. In our domain each patch has different height and width therefore, by taking $2m$ features for each patch the final feature vector become inconsistent for classification. To encounter this problem we padded zeros at the end of each smaller patch to make it fit and consistent for our classification model.

Figure 4.37 shows the results of local features. We used random forest model with upto 40 decision trees as we followed same in Bag of Features technique. As we observed from the figure 4.37 that we got 50.34% accuracy at tree 3. However, when we increased the number of decision trees the accuracy improved from 50.34% to 65.36% at tree 40. We increased 9% from tree 3 to 10. It means by increasing the decision tree from 3 to 40 we gained almost 15% accuracy which is better than BOF technique. The results in figure 4.37 are the average of 30 runs for each random forest model.

Figure 4.38 showing the 95% confidence interval for each random forest model. We can observed from the figure 4.38 that the SVD features for each random forest model have very narrow range or variance and none of them are overlapping. Therefore, we can say that we are 95% confident that each random forest model statistically not significant. It means 95% of times each random forest model for SVD features would have different overall score.

This technique is easy too implement and take less time for training and testing in random forest model. Hence, it produced much better results without involving clustering mechanism.

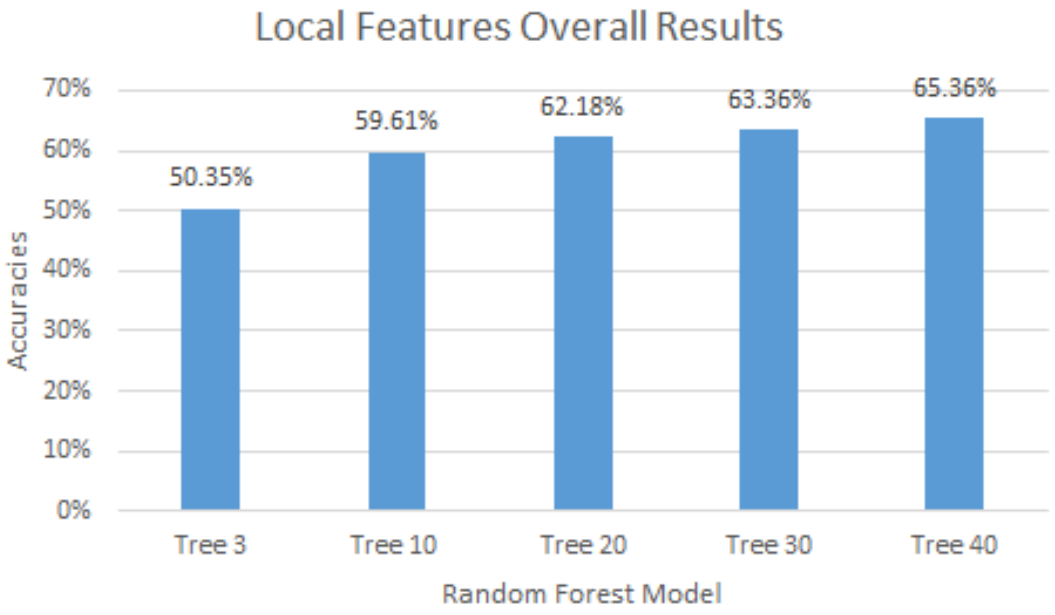


Figure 4.37: Local Features Overall Results

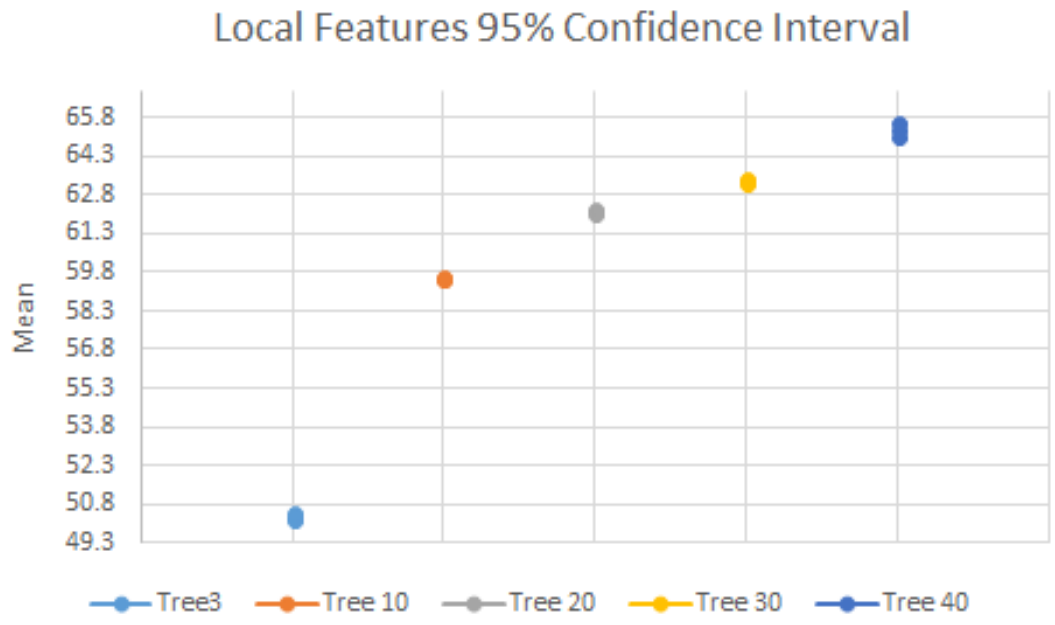


Figure 4.38: Local Features Average 95% Confidence Interval

Individual Body Parts

Table 4.17 represents the accuracies of 26 body parts using local features technique. Similar to other techniques in this technique we ran experiments up to 30 times and table 4.17 indicates the average accuracies of each body part for each random forest model.

In individual body parts results we got best ever results as compared to bag of features technique and as well as in pixel based techniques. Table 4.17 shows the results of individual body parts results from tree 3 to tree 40. Almost all the body parts scored over 50% ecxept one body part. However, in pixel based techniques the best we got only 4 body parts which scored below 50% in SIFT. We observed from table 4.17 that random forest model which consist of 40 trees is considered clear winner in terms of predicting maximum body parts with better score than other random forest models. The body parts "left_arm3" and "right_arm3" which their accuracies is less than 20% in all previous techniques. In this technique these body parts performed much better with over 50% accuracy. The highest accuracy is 87.48% for "right_head".

To conclude this we can say that the local features produced best individual body part accuracy as compared any other techniques. We improved upper body score but also improved lower body parts score. All the lower body parts score above 62% in which two lower body parts scored over 70%. Figure 4.39 shows the graphical representation of each body part accuracy.

Table 4.17: Local Features Average Individual Body Parts Results

Trees	right_head	right_face	left_face	neck	right_arm1	right_arm2	left_arm2	right_arm3	left_arm3
03	74.80%	58.96%	57.48%	69.24%	30.44%	34.14%	33.90%	43.12%	40.28%
10	81.99%	67.63%	64.11%	73.08%	41.53%	45.96%	42.60%	46.98%	47.46%
20	82.94%	69.69%	66.65%	74.30%	43.69%	49.35%	45.89%	47.91%	49.47%
30	84.53%	70.68%	67.05%	74.78%	45.68%	50.48%	47.39%	49.13%	50.53%
40	**87.48%**	**74.12%**	**70.22%**	**75.30%**	**51.03%**	**52.59%**	**50.12%**	**50.56%**	**52.88%**
Trees	**right_elbow**	**left_elbow**	**right_wrist**	**right_hand**	**right_chest**	**left_chest**	**right_abs**	**right_leg**	**right_foot**
03	32.85%	29.68%	33.09%	54.00%	69.81%	68.68%	53.23%	50.83%	45.65%
10	43.52%	39.26%	44.35%	65.54%	79.32%	79.97%	58.48%	58.27%	57.36%
20	46.76%	42.21%	48.29%	68.84%	82.11%	82.64%	61.47%	59.64%	61.31%
30	48.31%	43.98%	49.96%	70.10%	83.04%	83.63%	62.50%	60.64%	62.71%
40	**50.06%**	**46.47%**	**51.95%**	**72.45%**	**84.11%**	**84.81%**	**63.80%**	**61.37%**	**65.24%**
Trees	**thobe_part**	**left_head**	**left_arm1**	**left_wrist**	**left_hand**	**left_abs**	**left_leg**	**left_foot**	
03	66.77%	72.11%	38.93%	26.36%	50.88%	51.61%	50.49%	53.44%	
10	73.25%	80.81%	50.05%	40.90%	63.73%	60.18%	57.79%	64.35%	
20	75.13%	82.60%	52.76%	46.05%	67.16%	61.45%	60.30%	67.41%	
30	75.93%	83.31%	54.01%	48.03%	68.52%	62.05%	61.37%	68.51%	
40	**76.91%**	**86.17%**	**56.72%**	**51.47%**	**72.01%**	**62.79%**	**62.77%**	**70.35%**	

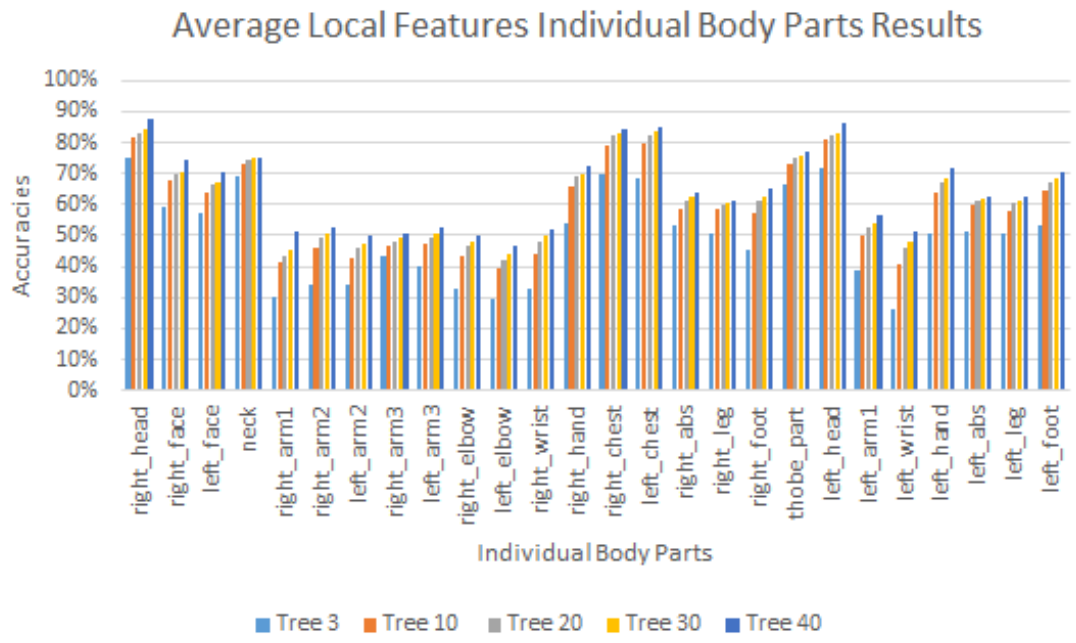


Figure 4.39: Local Features Average Individual Body Parts Results

Upper and Lower Body Parts Results

Figure 4.40 shows the upper and lower body parts scores for SVD technique. These average classes accuracies are calculated from confusion matrix using equation 4.3. Both upper and lower body parts scored over than 60%. This techniques scored much better than all previous techniques including pixel based techniques. Almost all the upper and lower body parts scored above 50% except one upper body part which is "left_elbow".

Figure 4.40a shows the average results for upper body using local features. The major improvement has been noticed in tree 10 where upper body parts improved almost 10% from tree 3. Similarly, from tree 10 to tree 40 we further improved 6% accuracy. The lowest upper body parts accuracy recorded 48.74% at tree 3 and highest is 64.15% at tree 40. In table 4.17 we observed that 9 body

parts scored between 50% and 60%. Interestingly, all these 9 body parts belongs to upper body. However, If we improved the accuracy of these body parts the average upper body parts can further be improved. These body parts are mostly small patches. The accuracy of small patches can be improved by adding some background pixels in the patch.

Figure 4.40b depicts the average lower body score. The average lower body scored better than upper body. This trend has been seen in almost every techniques except HOG where upper body part scored better. Similar to upper body we also noticed major improvement in lower body from tree 3 to tree 10 where we improved almost 9%. The lowest lower body parts accuracy recorded 53.44% at tree 3 and highest is 67.33% at tree 40. This is the best ever accuracy we got in this technique as compared to all other previous techniques. In table 4.17 we observed that all the lower body parts score above 60%.

Figure 4.41 shows the 95% confidence interval graph for upper and lower body parts. Figure 4.41a shows the 95% confidence interval graph for upper body parts. From tree 10 to tree 40 the confidence intervals are overlapping and their means are also appearing in the intervals so therefore, we can say that we are 95% confident that last four random forest model in figure 4.41a are similar or statistically significant. These random forest model producing almost same upper body score. This is similar to lower body 95% confidence interval as well in figure 4.41b. However, if we look at the interval of tree 3 and tree 10 in Figure 4.41a the interval of both random forest model are overlapping. But the mean of Tree 3

Table 4.18: Local Features Upper body Anova Test for Tree 3 and Tree 10

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.08952	1	0.08952	3.75693	0.05966	4.08474
Within Groups	0.95315	40	0.02382			
Total	1.04268	41				

is not appearing in tree 10 confidence interval. We don't know whether they are statistically significant or not. Therefore, we calculated the Anova test for both random forest model. The Hypothesis for Anova test are following:

- **Null Hypothesis 1**

$$\mu Tree3 = \mu Tree10 \tag{4.4}$$

- **Alternative Hypothesis 1**

$$\mu Tree3 < \mu Tree10 \tag{4.5}$$

The null hypothesis in equation 4.4 depicts that the upper body score for both random forest model are statistically significant. Whereas the alternative hypothesis in equation 4.5 depict that the upper score for random forest model are not statistically significant. We used 0.05 alpha value in Anova test.

Table 4.18 shows the results of Anova test for upper body. In table 4.18 we can see that F value is less than F-crit value so therefore, we failed to reject our null hypothesis in equation 4.4. It means that both random forest model are statistically significant.

Table 4.19: Local Features Lower body Anova Test for Tree 3 and Tree 10

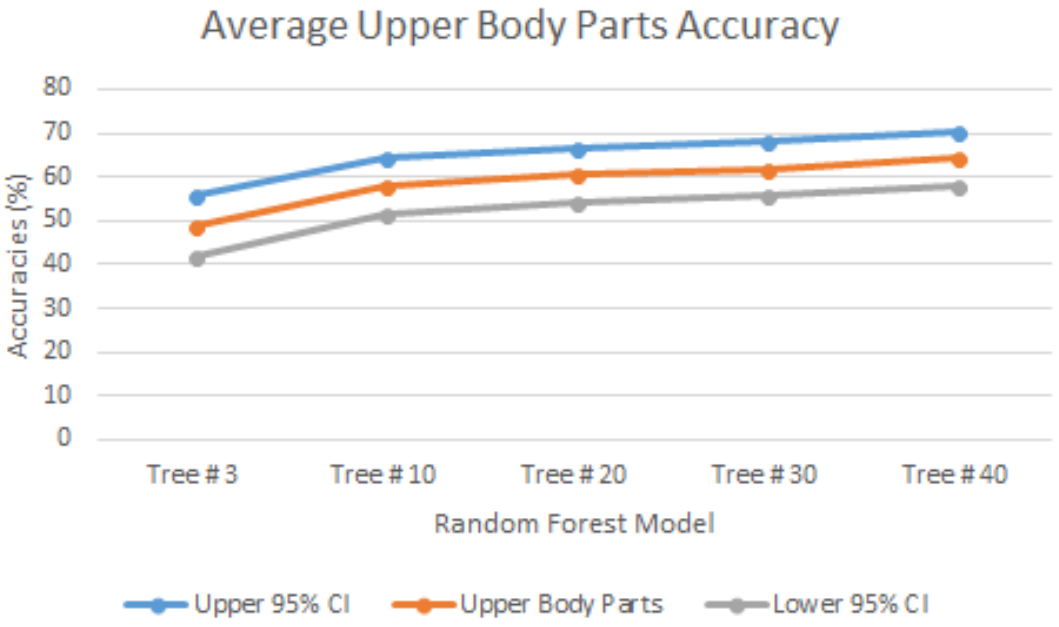
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.01921	1	0.01921	3.49969	0.09829	5.31765
Within Groups	0.04391	8	0.00548			
Total	0.06312	9				

In lower body parts we have a similar scenario like in upper body parts. The last four random forest models are statistically significant as we can see in the firgure 4.41b. However, in first two random forest models we are not sure that they statistically significant or not. Therefore, we also calculated Anova test for lower body as well. We used the same same hypothesis that we used for upper body in equation 4.4 and equation 4.5. Table 4.19 shows the results of Anova test for lower body. In table 4.19 we can see that F value is less than F-crit value so therefore, we failed to reject our null hypothesis in equation 4.4. It means that both random forest model are statistically significant with other random forest model.

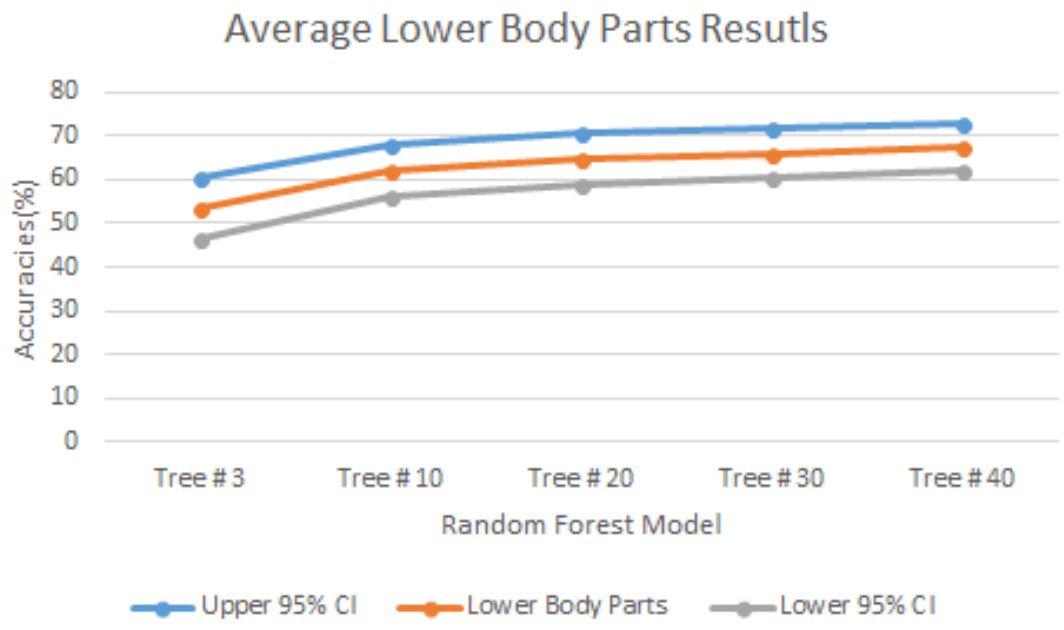
4.6 Draped Clothes Framework Overall Results and Discussions

4.6.1 Overall Results

Figure 4.42 represents the overall results of all the techniques that we used in draped clothes framework. We started with pixel based approach in which we

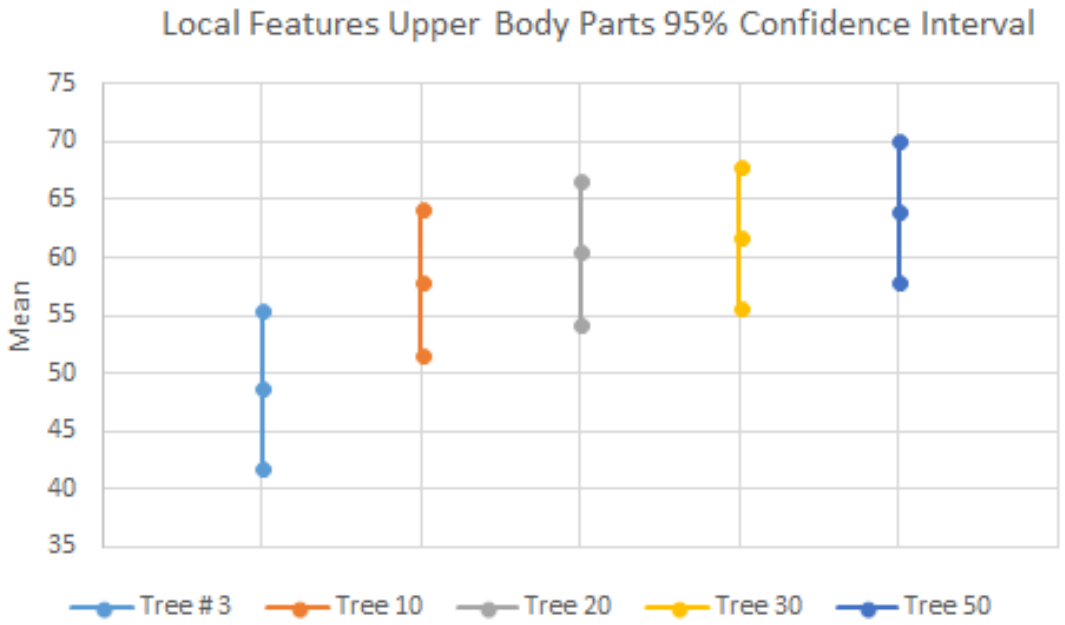


(a) Local Features Average Upper Body Parts Accuracies

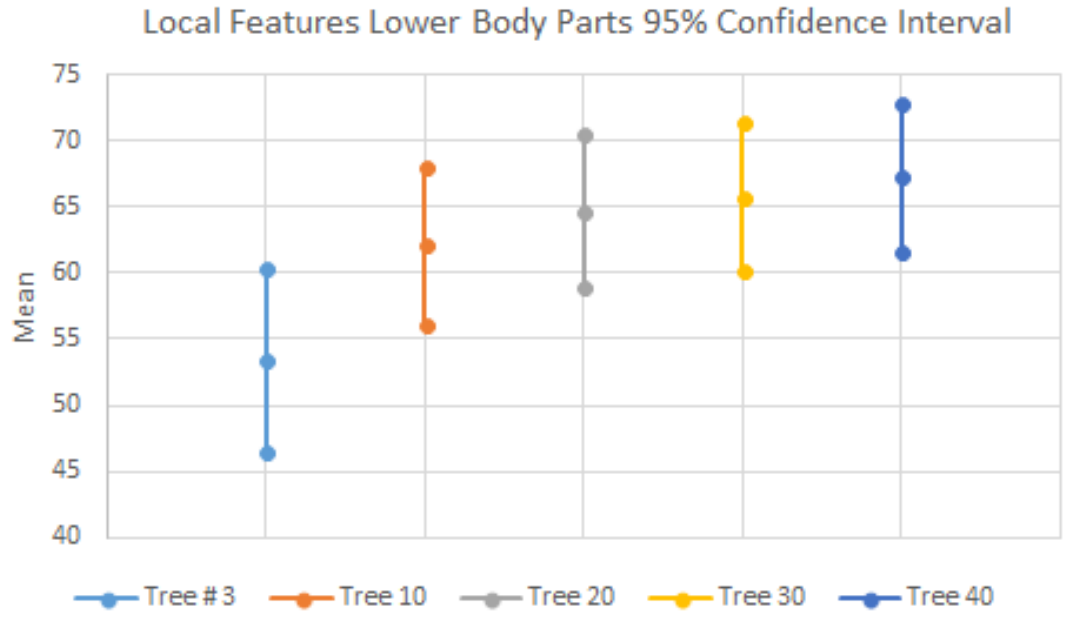


(b) Local Features Average Lower Body Parts Accuracies

Figure 4.40: Local Features Upper and Lower Body Parts Results



(a) Local Features Upper Body Parts 95% Confidence Interval



(b) Local Features Lower Body Parts 95% Confidence Interval

Figure 4.41: Local Features Upper and Lower Body Parts Confidence Interval Results

used depth and computer vision based approaches. BOF-SIFT technique has lowest score among other depth and computer vision techniques with 31% accuracy followed by MDF depth based technique with almost 44% accuracy. We used another depth techniques called LBP-DDF which produced better results than MDF and BOF-SIFT. Although, LBP-DDF beat low score techniques but still its accuracy is under 50% which is not very impressive. However, we then used computer vision techniques to improve the accuracy rate. We used HOG and SIFT feature descriptors. As we observed form the figure 4.42 SIFT descriptor is taking lead from other pixel based methods with 64% accuracy. It means majority of the instances were correctly classified. HOG scored almost 51% which is better than depth based feature but still the 49% of pixels or instances were wrong classified. We also used HOG-AVG technique which average the HOG window and considered average value as a feature. We achieved 44.26% accuracy which is similar to MDF depth based technique.

We used bag of feature and local features techniques in patch based approach. The BOF with HOG technique produced the same results that we achieved in pixel based approach using HOG. We got 51% of accuracy for BOF_HOG. Surprisingly, the SIFT in BOF never performed well as it performed in pixel based approach. It produced the lowest score among all other techniques with 31% accuracy. The reason of having low score is because, it may not able to cluster the patches correctly and considered two or more different patches in a same cluster. However, the local features technique outperformed not only patch based techniques but also

outperformed pixel based techniques. We achieved 65.36% accuracy which highest among all.

Figure 4.43 showing the 95% confidence interval for all techniques. We can observed from the figure 4.43 that all the techniques have very low variance. It is because of the higher population size. In our case the population size for each technique is 30. All the techniques have different confidence interval and no single technique is overlapping to each other. Hence, we can say that we are 95% confident that all techniques are statistically not significant to each other. It means every time we run the experiments of any techniques we will get different results than other techniques.

We can conclude from figure 4.42 in pixel based the SIFT produced much better results and local features in patch based produced almost the same results as we got in SIFT. In terms of computation and classification we advised to use local features for the detection human body parts in Arabic clothes. Because it is easy to implement and take very less time in execution. On the other side the SIFT take more time to execute and also take lot of time for training. The result of SIFT in figure 4.42 was classified only on 3 tree and with 800 random pixels. However, by increasing the number of trees and random pixels may increase the results of SIFT.

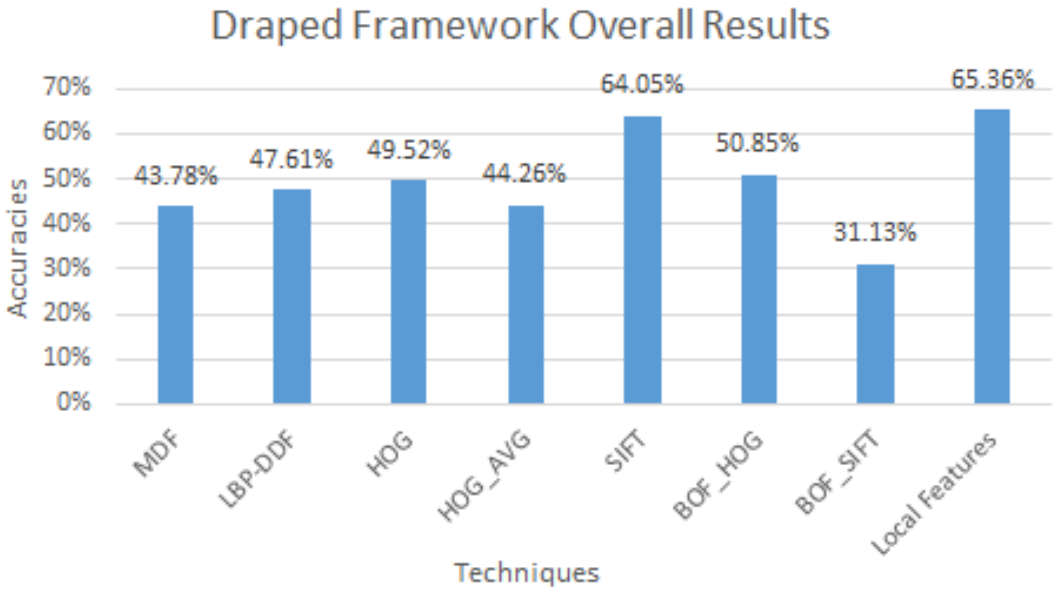


Figure 4.42: Draped Clothes Framework Overall Results

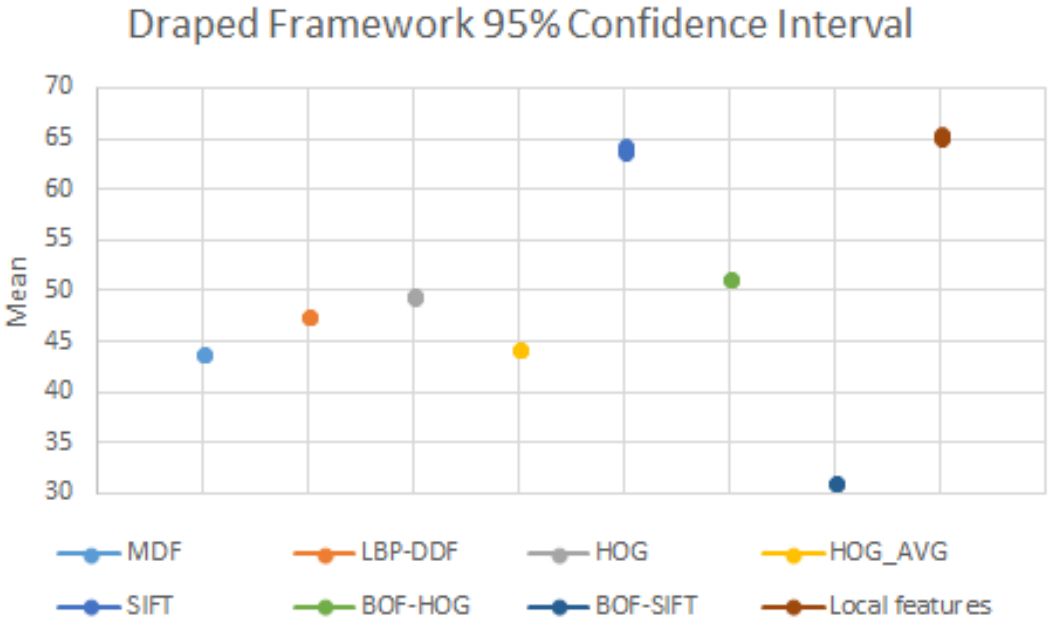


Figure 4.43: Draped Clothes Framework Confidence Intervals for All Techniques

4.6.2 Overall Individual Body Parts Results

Table 4.20 shows the individual body parts results for all pixel and patch based techniques. The accuracies of each body part in every technique is the average of 30 runs. It is clear from the Table 4.20 that both SIFT and local features are clear winner in terms of predicting maximum body parts. The BOF-SIFT performed worst among all other techniques followed by HOG-Average technique. However, we noticed huge improvement in the accuracy when we compared SIFT and Local features individual body parts accuracies with the rest of the techniques.

The "right_head" body part got highest score among all other body parts with almost 87.48% accuracy followed by "left_head" body part with accuracy of 86.17% using local features technique. The reason of high accuracies of these body parts depends on couple of important factors. First, in our dataset the position of these two body parts are almost fixed in every image frame which is on the top of the image. These body parts never appeared over any body parts which avoid the chance of resemblance between two or more body parts of having a same feature. Second, the involvement of background pixels. It was noticed from MDF technique that the background pixels helps body parts to increase their accuracy. However, there are other 7 body parts including above two body parts which scored over 50% in almost every pixel and patch based technique except BOF-SIFT. This means that these body parts have dominant features which give us a correct location of desired body parts more than 50% accuracy. The body parts are "right_face", "left_face", "left_head", "right_head", "right_chest", "right_abs"

and “thobe_part”.

The “left_ankle” body parts performed worst in pixel based technique. This body part was ignored in patch based techniques because of having a very small patch size and it is rarely appeared on the image frame. We improved almost 20% accuracy in SIFT than other pixel based techniques. Similarly, the body part “right_arm3” also performed worst in pixel based techniques. The worst we got 2% in HOG and 15% in SIFT. This similar situation observed in other body parts as well like “left_arm3”, “left_wrist” and “right_arm3”. The reason is that these body parts are very small in size, sometimes are occluded by a body pose and sometimes it appears over another body parts that made resemblance between two body parts by a having a same features values. The other reason is the absence of background pixels most of times specifically when these body parts appeared over another body parts. In this case our defined offset distance unable to cover background pixels. However, when we switch to patch based technique from pixel based technique we observed noticeable improvement in all smaller body parts. The huge improvement was recorded on body part “right_arm3” where we improved more than 35% accuracy in local features. All the smaller body parts scored over 50%.

One possible solution to improve the performance these smaller body parts by extending the offset distance. Increasing in the offset distance able to cover background pixels while computing features. We already know the background pixels help to improve the accuracy of body parts. The other possible solution is

to merge two smaller parts together and considered it as an one part for example merge"left_arm2" with "left_arm3". This will increase the patch size and we would have more random pixels hit in pixel based technique to encounter more features. Similarly, in patch based technique the solution would worked well because we noticed bigger patch have over 60% accuracy than smaller one. Therefore, we can improve the accuracies of these body parts in patch based technique as well.

All the lower body parts scored over 60% except body part "left_ankle". The "thobe_part" made first appearance in the list of high accuracies of lower body with 76.91% of accuracy followed by "left_foot" body parts with 70.35% accuracy. The one of the main purpose of this thesis to improve the accuracy performance of lower body parts as compared to previous work [3] which we did successfully using SIFT and local feature techniques. Out of total 27 body parts the local feature technique topped on the table for having maximum accuracies of 16 body parts. The SIFT technique in Pixel based approach predicted 11 body parts with maximum accuracies.

In conclusion to individual body parts results we would say that local features technique in patch based approach outclass other techniques for predicting maximum number of body parts with high accuracies followed by SIFT. Figure 4.44 shows the graphical representation of each body part accuracy.

Table 4.20: Draped Framework Individual Body Parts for All Techniques

	left_arm1	**left_arm2**	**left_arm3**	**right_abs**	**left_elbow**	**right_leg**	**thobe_part**	**left_abs**	**left_leg**
MDF	30.15%	35.50%	17.17%	52.86%	36.17%	42.86%	50.01%	42.12%	43.09%
DDF	32.31%	39.44%	23.84%	55.10%	43.97%	45.77%	51.38%	46.95%	45.42%
HOG	36.79%	39.50%	18.91%	54.15%	48.73%	45.10%	51.67%	48.92%	46.26%
H_AVG	29.23%	35.32%	17.61%	53.28%	36.68%	43.37%	50.03%	43.19%	43.39%
SIFT	**58.66%**	**60.09%**	37.79%	**67.92%**	**60.02%**	56.71%	62.99%	**64.88%**	60.11%
B_HOG	46.17%	39.26%	35.17%	59.28%	37.16%	50.00%	70.04%	55.77%	50.51%
B_SIFT	34.50%	23.27%	22.45%	22.52%	21.50%	22.11%	44.38%	26.46%	20.26%
LF	56.72%	50.12%	**52.88%**	63.80%	46.47%	**61.37%**	**76.91%**	62.79%	**62.77%**
	left_chest	**right_chest**	**right_arm1**	**right_arm2**	**right_arm3**	**right_elbow**	**left_face**	**right_face**	**neck**
MDF	41.24%	50.44%	25.58%	29.91%	7.34%	36.97%	49.21%	52.33%	38.45%
DDF	46.44%	53.31%	29.54%	35.74%	9.66%	44.06%	56.62%	58.36%	53.20%
HOG	53.78%	60.17%	28.74%	37.02%	2.00%	45.36%	66.21%	63.77%	51.41%
H_AVG	42.25%	50.73%	26.97%	29.94%	8.18%	36.83%	48.22%	52.79%	36.68%
SIFT	73.14%	74.53%	**60.06%**	**60.74%**	15.15%	**57.58%**	**80.66%**	**80.07%**	60.88%
B_HOG	84.03%	83.42%	43.12%	40.76%	38.00%	43.80%	62.75%	62.52%	63.42%
B_SIFT	27.88%	30.64%	33.10%	25.30%	14.99%	16.54%	50.01%	43.88%	73.42%
LF	**84.81%**	**84.11%**	51.03%	52.59%	**50.56%**	50.06%	70.22%	74.12%	**75.30%**
	right_foot	**left_head**	**right_head**	**right_wrist**	**right_hand**	**left_foot**	**left_hand**	**left_wrist**	**left_ankle**
MDF	44.89%	53.54%	54.47%	28.42%	29.88%	47.68%	30.74%	25.45%	8.59%
DDF	51.42%	57.69%	57.63%	35.33%	39.40%	53.95%	38.60%	32.96%	12.26%
HOG	55.76%	60.50%	60.51%	37.87%	54.03%	56.41%	51.67%	27.21%	8.86%
H_AVG	43.97%	49.29%	52.30%	28.65%	30.13%	46.19%	32.78%	26.37%	11.24%
SIFT	63.69%	78.10%	79.10%	50.16%	50.89%	63.78%	47.60%	41.78%	**36.15%**
B_HOG	46.01%	56.72%	55.26%	31.28%	52.14%	49.08%	47.31%	31.53%	NA
B_SIFT	29.68%	48.60%	47.71%	26.98%	25.12%	33.92%	23.51%	23.25%	NA
LF	**65.24%**	**86.17%**	**87.48%**	**51.95%**	**72.45%**	**70.35%**	**72.01%**	**51.47%**	NA

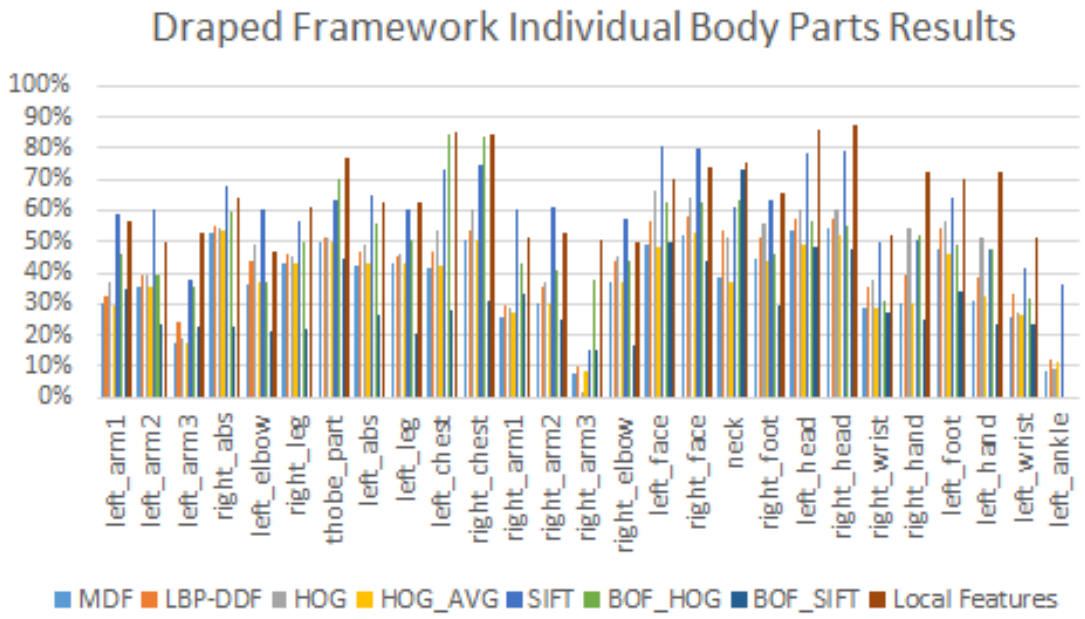


Figure 4.44: Individual body parts overall Accuracies

4.6.3 Overall Upper and Lower Body Parts Results

Figure 4.45 shows the accuracies of lower and upper body parts for all pixel and patch based methods. Likewise, in Individual body parts here also the local feature and SIFT techniques are clear winners of having maximum average upper and lower body accuracy as compared to others techniques. The local feature technique performed well in both upper and lower body parts followed by SIFT. We achieved maximum average lower body part score of 67% in local features. However, the accuracy of average upper body is 64% . Although, LBP-DDF technique is better than MDF technique. But, we improved almost 20% accuracy using SIFT and local feature technique from MDF and other pixel based techniques.

The BOF using SIFT have worst average upper and lower body score followed by MDF technique. HOG_AVG performed worst then actual HOG in both upper

and lower body parts. However, HOG_AVG technique has better lower body score than upper body. Whereas, in HOG technique the upper body is leading. Comparing, HOG with BOF-HOG we figure it out by involving clustering using HOG gave us better individual body parts score, better average upper body parts score and better average lower body parts score. However, we found opposite case in terms of comparing SIFT in pixel based with BOF-SIFT in patch based technique. The BOF-SIFT performed not only worst than SIFT but also performed worst than other technique. We figure it out that by applying clusters using SIFT the different patches were not differentiated properly. This is the reason the BOF-SIFT performed worst than all. One possible solution is to use a RGB dataset instead of depth values dataset. The values in depth dataset are distance from the camera and that's why the same patches were clustered properly.

Total out of 8 techniques only three techniques able to produce more 50% average upper and lower body score. In which two of the techniques are from patch based methods. Therefore, we can conclude that the depth features are not well enough to produced better average upper and lower body accuracy as compared computer vision feature techniques.

4.6.4 Random Pixel Analysis

We used different number of random pixels such as 800, 1000, 1200, 1400 and 2500 pixels. We decided to analyze these pixels information and check how many average pixel each part is taking in one image. Figure 4.46 shows the average

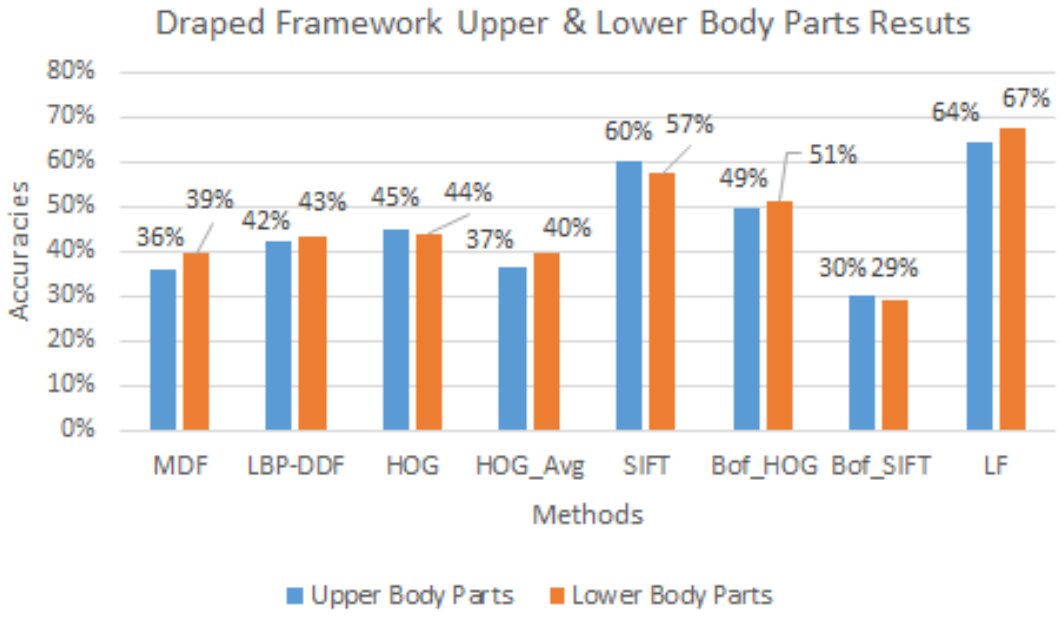


Figure 4.45: Lower and Upper body parts overall Accuracies

random pixel by each part in one image.

We observed from figure 4.46 that lower body parts taking maximum number of pixels as compared to upper body parts. We also observed that body part "left_ankle" taking very minimum number of pixels from other body part. The reason of getting low pixels because we observed the dataset and we found this body part is rarely visible in the image. Most of the time this body part is covered by a thobe dress.

4.6.5 Comparison with Existing Work

Figure 4.47 stated the comparison of our work with existing work. As we can notice from the figure 4.47 the SIFT and local feature technique outclass all other methods. Our median depth feature and decimal depth feature performed slightly

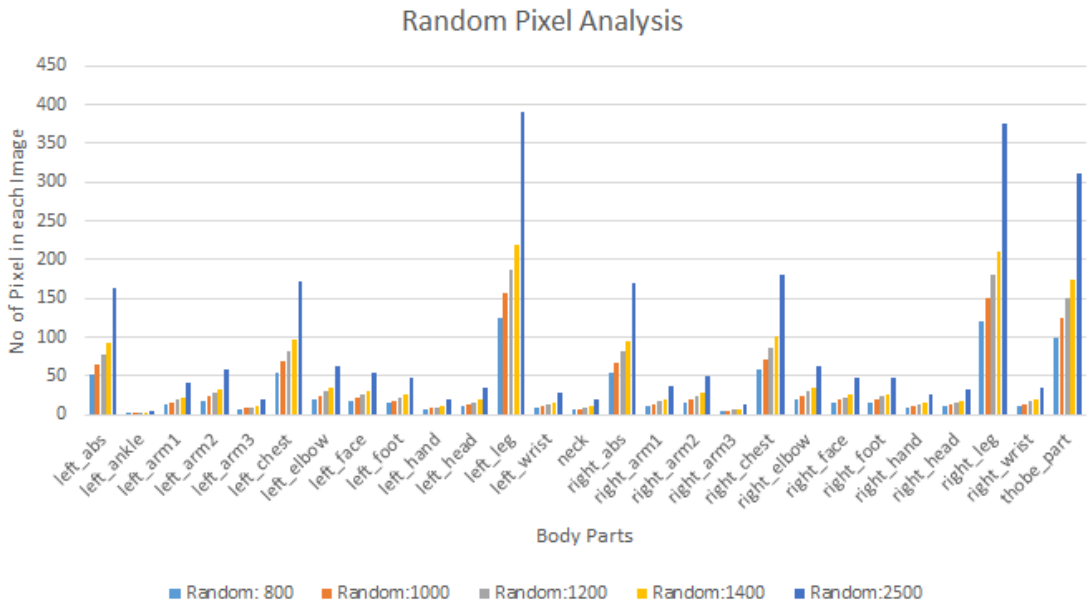


Figure 4.46: Average number of random pixel by each part

better than Shotton *et al.* [2] and Ridwan [3] feature. However, both SIFT local feature technique improved more than 20% accuracy performance as compared to Shotton *et al.* [2] and Ridwan [3]. whereas HOG improved 10%. The LBP-DDF and fusion techniques also performed slightly better than Shotton *et al.* [2] and Ridwan [3].. The BOF using SIFT performed worst than all other technique including Shotton *et al.* [2] and Ridwan [3].

Therefore, we can say that computer vision feature techniques performed much better than simple depth features.

4.6.6 Overall Body Parts Accuracy Distribution

Table 4.21 shows the body parts accuracy distribution for each pixel and patch based technique. We sorted the accuracy distribution from worst(top) to best(bottom) in table 4.21. The purpose is to show how we improved individ-

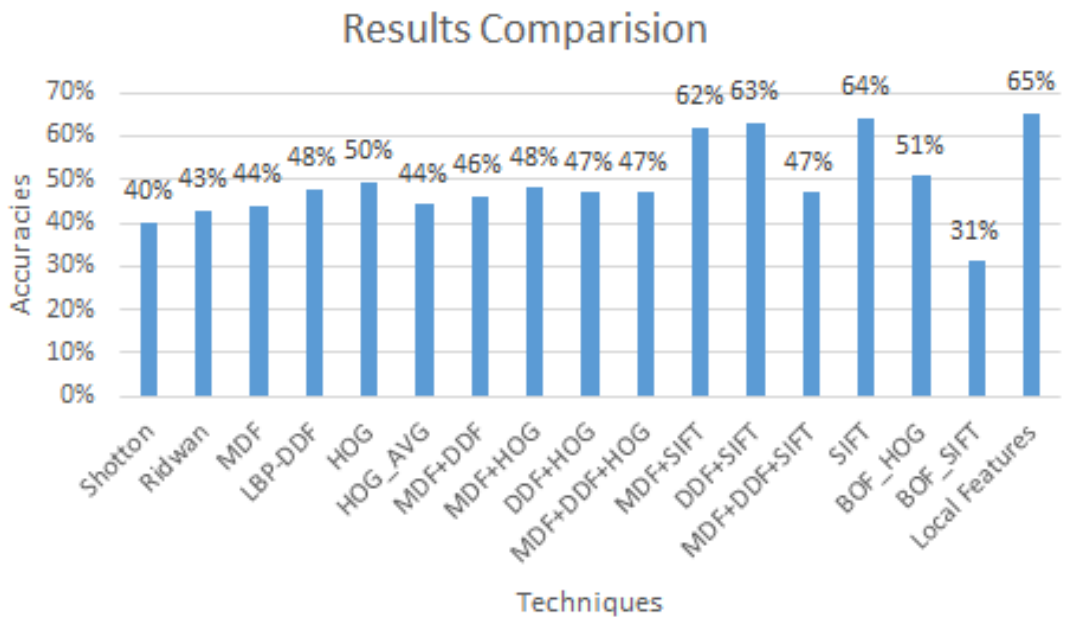


Figure 4.47: Comparison with existing work

ual body parts from worst to better by applying different pixel and patch based techniques. The BOF-SIFT has worst body parts accuracy distribution and local features(LF) has best body parts accuracy distribution as described in table 4.21.

As we already know that BOF-SIFT technique performed worst than all and it is clearly reflected from table 4.21. Only 2 body parts scored above 50% both parts belongs to upper body("left_face" and "neck"). Second technique we used was HOG-AVG. This technique taking the average of actual HOG window and considered as a feature for random classified pixel. We improved from BOF-SIFT however, only 5 body parts scored above 50% and all these parts lies between 50% and 60% slot. Majority of body Parts belongs to upper body. The third and fourth techniques were related to simple depth features. However, we further improved 1 and 4 body parts respectively from first two techniques. But still no body parts scored above 60%. After getting no major improvements from

simple depth features based techniques. We used HOG this technique considered very useful in human and pedestrian detection. We improved more body parts as compared to above techniques in table 4.21 using HOG. Now total 13 body parts scored above 50% in which 5 body parts scored above 60% and none of the body parts scored above 70%. Out of 13 body parts 10 body parts belongs to upper body and rest of belongs to lower body parts.

The BOF-HOG patch based technique overall made third position for predicting most body parts. The BOF-HOG is patch based technique so we have total 26 body parts instead of 27 because body part "left_ankle" is ignored due to small patch size. The BOF-HOG scored 13 body parts above 50% in which 7 body parts scored within 50% to 60%, 3 body parts scored within 60% to 70%, 1 body parts scored within 70% to 80% and 2 body parts scored above 80%. However, HOG in pixel based also scored 13 body parts above 50% but BOF-HOG technique has better accuracy distribution as mentioned in table 4.21.

The last two techniques in table 4.21 outperformed other techniques by having a much better accuracy distribution. SIFT in pixel based technique outperformed all other pixel based technique. Out of 27 body parts the SIFT scored only 5 body parts below 50% and rest of them scored above 50%. In which 11 body parts scored within 60% to 70%, 4 body parts scored within 70% to 80% and 2 body parts scored above 80%. However, the local features technique performed better than SIFT by having only 1 body part scored below 50% and the rest of the body parts scored above 50%. Out of 26 body parts the local feature technique

Table 4.21: Body Parts Accuracy Distribution

	Below 50%	50% - 60%	60% - 70%	70% - 80%	Above 80%
B_SIFT	24	1	0	1	0
H_AVG	22	5	0	0	0
MDF	21	6	0	0	0
DDF	17	10	0	0	0
HOG	14	8	5	0	0
B_HOG	13	7	3	1	2
SIFT	5	5	11	4	2
LF	1	9	5	7	4

scored 16 body parts above 60% in which 4 body parts scored above 80%, 7 body parts scored within 70% to 80% and 5 body parts scored with in 60% to 70%. All the lower body parts scored above than 60% in local features technique. Whereas, in SIFT only 4 lower body parts out of 6 scored above 60%.

In conclusion, we started with 24 body parts which scored below 50% and we end up with only 1 body part which scored below 50% in local feature technique. This is a significant improvements as compared to other techniques.

4.6.7 Draped Clothes Framework Final Output

As we discussed earlier in draped clothes framework we applied different feature extraction techniques for recognizing human body parts. Our framework would show the best technique in both pixel and patch based approach. Figure 4.48 shows our framework final output. In pixel based approach we got SIFT that gave us 64% accuracy which is best among all other techniques we used. Where as in Patch based approach we used different techniques such as Bag of Features and local features. However, in patch based technique the local features technique

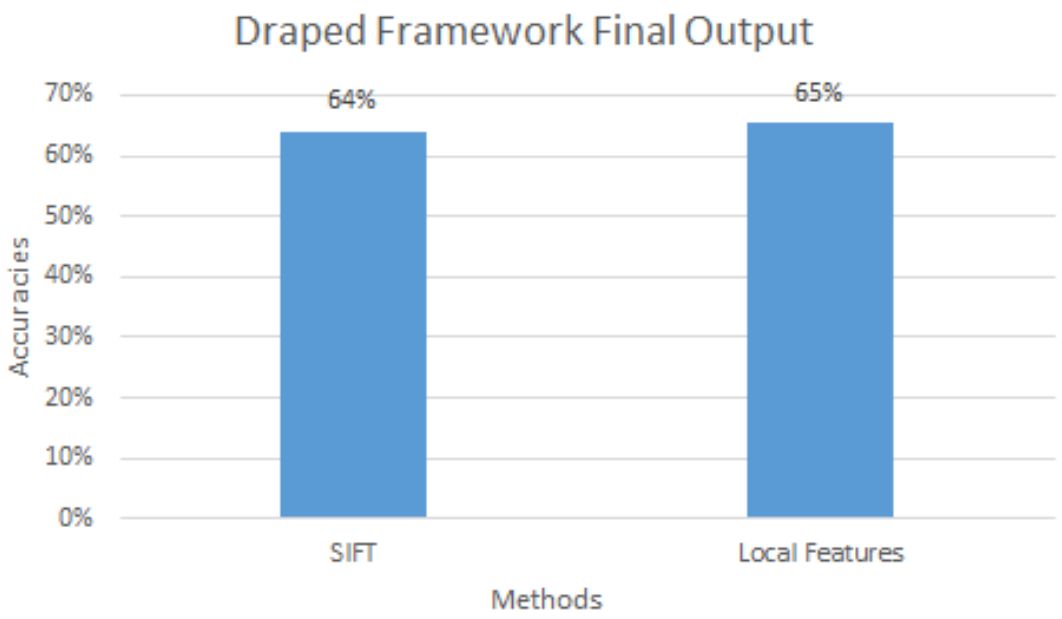


Figure 4.48: Draped Framework Output

topped with 65% accuracy.

To conclude that our draped clothes based framework figure out that SIFT and local features performed better as compared to other techniques. However, the local features gives better individual and upper lower body parts accuracy as compared to SIFT. The local features are easy to implement and take less time in classification as compared to SIFT in pixel based approach.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Human pose recognition is considered a well-known process of estimating the human body pose from a single image or from video frames. It is a kind of part based computer vision problem. The body parts are recognized from the whole body and using these recognized parts the exact human pose can be predicted. In human pose recognition, most of the research has been taken on western clothes since this problem first occurred. The structure of western clothes is very simple. The fabric is not covering the entire body. Some of the body parts are separated to each other for example left arm, right arm, left leg and right leg. That is why human wearing jeans, casual and dress shirts or sports trousers it is easy recognized upper and lower body parts using low-cost depth cameras.

However, there is no such research exists that recognized human pose on draped clothes. The draped clothes such as Arabic Thobe and subcontinent dresses are difficult to recognized human pose. These clothes are unlike western clothes it covers the entire body with a single piece of fabric. Therefore, in such cases the lower body would look like a concrete opaque square in a 2D image hiding all the spatial details of lower body parts left thighs, right thighs, left knees, right knees, left leg and right leg. Even the low-cost depth cameras are failed to detect these body parts.

Therefore, in this thesis, we present a framework called draped clothes framework that recognized human pose in draped based clothes. In this framework we adopted learning based technique in Pixel and Patch based methods for recognizing human pose. In each method we applied depth and computer vision feature extraction techniques. We used random forest classifier for the classification human body parts. After applying all the techniques in both methods. Our draped clothes framework figure it out that in pixel based approach the SIFT technique outclass other feature techniques leading with 64% accuracy. We improved more than 20% of accuracy from the existing work. We also improved the lower body part accuracy overall up to 57%. While in patch based method we used Bag of feature technique with SIFT and HOG and local feature technique. We found that local feature is very useful for predicting the correct body parts with almost 65% accuracy. The local feature technique have better individual body part accuracy rate and also have better average lower body parts accuracy with upto 67%.

The local feature technique predicted more body parts with higher accuracy than SIFT.

5.2 Threats to Validity

5.2.1 Construct Validity

Every research have some limitation. One of the limitation in our thesis is the dataset. We used synthetic dataset. which is modeled by Ridwan [3] in Maya. There is no real dataset available publicly. The real dataset is different from synthetic dataset in so many ways. For example in real dataset the main factor is noise in the image which the synthetic dataset lacked.

The other limitation is null feature vector at random pixel. We used SIFT and HOG techniques which is calculate a gradients in a 8x8 window. If the windows lies outside the image the feature at that points returned null vector. It means that we loosed some the random pixels features. These random pixels are mostly the border pixels. We cannot include null values in final feature vector. Because it act as missing values in the classifier and can affect the accuracy as well. That's why we removed null vector during classification.

The last limitation in our thesis is related to random pixel as well. We are using different random pixel it may be the chance that two or more pixel may point to the same pixel. So therefore, we may have a repetition of same features. Also the some body parts may get low number of random spot and some high.

5.2.2 External Validity

In our thesis all the patch and pixel based implementation is done on Matlab 2017 [80] that can restrict our result to Matlab only. We also use KNIME analytics tool [81] for the classification purposes. Errors in KNIME analytics tool can make difference in our results.

5.3 Future Work

In future, we would like to do following.

- We will extend our work to further improve the human body part detection accuracy.
- We will create new dataset which consist of real thobe images.
- We will also test our methods with other draped clothes dataset.
- In thesis we restricted our self to he recognition of human body parts in Arabic thobe. But we will also work to recognize the human pose and action as well.
- We will also overcome the limitation of our current research work.

REFERENCES

[1] "Sensor survey." http://rosindustrial.org/news/2016/1/13/3d-camera-survey, accessed: 2017-01-10.

[2] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[3] M. A. Ridwan Jalali, Adel Ahmed and L. Gouti, "Pose estimation of human wearing thobe using depth images," *MS Thesis*, 2016.

[4] H. Abe, "Microsoft kinect depth sensor internal structure details," 2017, accessed: 2017-10-24. [Online]. Available: http://www.aud.ucla.edu/programs/m_arch_ii_degree_1/studios/2013_2014/gehry/?p=786.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[6] T. Watanabe, S. Ito, and K. Yokoi, "Co-occurrence histograms of oriented gradients for pedestrian detection," *Advances in Image and Video Technology*, pp. 37–47, 2009.

[7] Y. Meng, B. Tiddeman *et al.*, "Implementing the scale invariant feature transform (sift) method," *Department of Computer Science University of St. Andrews*, 2006.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning [internet]," *Microsoft Research*, 2011.

[10] Z. Liu, J. Zhu, J. Bu, and C. Chen, "A survey of human pose estimation: the body parts parsing based methods," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 10–19, 2015.

[11] H. Aviezer, Y. Trope, and A. Todorov, "Body cues, not facial expressions, discriminate between intense positive and negative emotions," *Science*, vol. 338, no. 6111, pp. 1225–1229, 2012.

[12] R. L. Birdwhistell, *Kinesics and context: Essays on body motion communication*. University of Pennsylvania press, 2010.

[13] D. Ramanan, "Learning to parse images of articulated bodies," in *Advances in neural information processing systems*, 2007, pp. 1129–1136.

[14] A. Hernández-Vela, N. Zlateva, A. Marinov, M. Reyes, P. Radeva, D. Dimov, and S. Escalera, "Graph cuts optimization for multi-limb human segmentation in depth maps," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 726–732.

[15] Z. Zhang, Y. Liu, A. Li, and M. Wang, "A novel method for user-defined human posture recognition using kinect," in *Image and Signal Processing (CISP), 2014 7th International Congress on.* IEEE, 2014, pp. 736–740.

[16] C. Youness and M. Abdelhak, "Machine learning for real time poses classification using kinect skeleton data," in *Computer Graphics, Imaging and Visualization (CGiV), 2016 13th International Conference on.* IEEE, 2016, pp. 307–311.

[17] I. Behoora and C. S. Tucker, "Machine learning classification of design team members' body language patterns for real time emotional state detection," *Design Studies*, vol. 39, pp. 100–127, 2015.

[18] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[19] B. Holt, E.-J. Ong, H. Cooper, and R. Bowden, "Putting the pieces together: Connected poselets for human pose estimation," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1196–1201.

[20] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1385–1392.

[21] M. Sun and S. Savarese, "Articulated part-based model for joint object detection and pose estimation," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 723–730.

[22] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 755–762.

[23] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3810–3818.

[24] M. Jiu, C. Wolf, G. Taylor, and A. Baskurt, "Human body part estimation from depth images via spatially-constrained deep learning," *Pattern Recognition Letters*, vol. 50, pp. 122–129, 2014.

[25] E. A. Suma, B. Lange, A. S. Rizzo, D. M. Krum, and M. Bolas, "Faast: The flexible action and articulated skeleton toolkit," in *Virtual Reality Conference (VR), 2011 IEEE*. IEEE, 2011, pp. 247–248.

[26] J.-w. Kang, D.-j. Seo, and D.-s. Jung, "A study on the control method of 3-dimensional space application using kinect system," *International Journal of Computer Science and Network Security*, vol. 11, no. 9, pp. 55–59, 2011.

[27] N. Nguyen-Duc-Thanh, D. Stonier, S. Lee, and D.-H. Kim, "A new approach for human-robot interaction using human body language," *Convergence and Hybrid Information Technology*, pp. 762–769, 2011.

[28] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[29] Y.-J. Chang, S.-F. Chen, and J.-D. Huang, "A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities," *Research in developmental disabilities*, vol. 32, no. 6, pp. 2566–2570, 2011.

[30] L. Xia, C.-C. Chen, and J. K. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on.* IEEE, 2011, pp. 15–22.

[31] M. Siddiqui and G. Medioni, "Human pose estimation from a single view point, real-time range sensor," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on.* IEEE, 2010, pp. 1–8.

[32] R. Z. L. Hu, "Vision-based observation models for lower limb 3d tracking with a moving platform," 2011.

[33] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 1014–1021.

[34] M. Eichner and V. Ferrari, "We are family: Joint pose estimation of multiple persons," in *European Conference on Computer Vision.* Springer, 2010, pp. 228–242.

[35] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari, "2d articulated human pose estimation and retrieval in (almost) unconstrained still images," *International journal of computer vision*, vol. 99, no. 2, pp. 190–214, 2012.

[36] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973.

[37] M. Sanzari, V. Ntouskos, and F. Pirri, "Bayesian image based 3d pose estimation," in *European Conference on Computer Vision.* Springer, 2016, pp. 566–582.

[38] T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Ieee iccv*, vol. 99, no. 1999, 1999, pp. 1–19.

[39] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999, pp. 246–252.

[40] P. D. Z. Varcheie, M. Sills-Lavoie, and G.-A. Bilodeau, "A multiscale region-based motion detection and background subtraction algorithm," *Sensors*, vol. 10, no. 2, pp. 1041–1061, 2010.

[41] C. Guillot, M. Taron, P. Sayd, Q. C. Pham, C. Tilmant, and J.-M. Lavest, "Background subtraction adapted to ptz cameras by keypoint density estimation," in *Proceedings of the British Machine Vision Conference*, 2010, pp. 34–1.

[42] F. Wang and Y. Li, "Beyond physical connections: Tree models in human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 596–603.

[43] A. Fathi and G. Mori, "Human pose estimation using motion exemplars," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.* IEEE, 2007, pp. 1–8.

[44] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[45] B. Sapp, D. Weiss, and B. Taskar, "Parsing human motion with stretchable models," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1281–1288.

[46] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1-2, p. 4, 2010.

[47] "Eval dataset." http://www.comp.leeds.ac.uk/mat4saj/lsp.html, accessed: 2017-01-10.

[48] "Lsp dataset." http://www.comp.leeds.ac.uk/mat4saj/lsp.html, accessed: 2017-01-10.

[49] "Parse dataset." https://computing.ece.vt.edu/ santol/projects/zsl_via_visual_abstraction/parse/index.html, accessed: 2017-01-10.

[50] T. Helten, "Processing and tracking human motions using optical, inertial, and depth sensors," 2013.

[51] "Flic dataset." http://bensapp.github.io/flic-dataset.html, accessed: 2017-01-10.

[52] "Pascal dataset." http://host.robots.ox.ac.uk/pascal/VOC/, accessed: 2017-01-10.

[53] "Buffy dataset." http://www.robots.ox.ac.uk/ vgg/data/stickmen/, accessed: 2017-01-10.

[54] "Mpii dataset." http://human-pose.mpi-inf.mpg.de/, accessed: 2017-01-10.

[55] "Mixing body-part sequences for human pose estimation dataset." http://lear.inrialpes.fr/research/ posesinthewild/, accessed: 2017-01-10.

[56] "Human 3.6h dataset." http://vision.imar.ro/human3.6m/description.php, accessed: 2017-01-10.

[57] "Cmu-mocap dataset." http://mocap.cs.cmu.edu/, accessed: 2017-01-10.

[58] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 1194–1201.

[59] "Umpm benchmark: A multi-person dataset," https://www.projects.science.uu.nl/umpm/, accessed: 2017-01-10.

[60] "Tum kitchen dataset," https://ias.cs.tum.edu/software/kitchen-activity-data., accessed: 2017-01-10.

[61] "Kth multiview football dataset." http://www.csc.kth.se/cvap/cvg/?page=software, accessed: 2017-01-10.

[62] "Video pose dataset." http://bensapp.github.io/videopose-dataset.html, accessed: 2017-01-10.

[63] B. Waldvogel, "Accelerating random forests on cpus and gpus for object-class image segmentation," *Master's Thesis of Rheinische Friedrich Wilhelms Universitt Bonn*, 2013.

[64] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[65] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[66] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[67] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," in *Readings in Computer Vision*. Elsevier, 1987, pp. 671–679.

[68] J. L. Crowley and R. M. Stern, "Fast computation of the difference of low-pass transform," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 212–222, 1984.

[69] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[70] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[71] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[72] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[73] S. Nowozin, "Improved information gain estimates for decision tree induction," *arXiv preprint arXiv:1206.4620*, 2012.

[74] V. Y. Kulkarni and P. K. Sinha, "Random forest classifiers: a survey and future research directions," *Int J Adv Comput*, vol. 36, no. 1, pp. 1144–53, 2013.

[75] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[76] P. Yin, A. Criminisi, J. Winn, and I. Essa, "Tree-based classifiers for bilayer video segmentation," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[77] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[78] A. Criminisi, J. Shotton, and S. Bucciarelli, "Decision forests with long-range spatial context for organ localization in ct volumes," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009, pp. 69–80.

[79] "The vlfeat open source library implements popular computer vision algorithms." http://www.vlfeat.org/overview/sift.html., accessed: 2017-10-01.

[80] "Matlab programming tool." www.mathworks.com/products/matlab.html., accessed: 2018-02-01.

[81] "Knime analytics platform." https://www.knime.com., accessed: 2018-02-01.

Vitae

- Name: Faisal Sajjad
- Nationality: Pakistani
- Date of Birth: June 25, 1991
- Email: *fsajjad342@gmail.com*
- Permenant Address: Islamabad, Pakistan
- Zip/Postal Code: 44000
- Thesis Publications 1: Faisal Sajjad, Adel F. Ahmed, Moataz A. Ahmed. "A study on the learning based human pose recognition" Solutions for a Smarter Economy, 2017 9th IEEE GCC conference & exhibition.
- Thesis Publications 2: Faisal Sajjad, Adel F. Ahmed, Moataz A. Ahmed, Lahouari Ghouti. "Draped clothes based human pose recognition using depth images" Journal of computer vision and image understanding *[To be submitted]*.